

UNIVERSITATEA POLITEHNICA BUCUREȘTI
FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE
DEPARTAMENTUL CALCULATOARE



PROIECT DE DIPLOMĂ

Aplicație web pentru managementul învățării

Panait Ștefan-Andrei

Coordonator științific:

Prof. dr. ing. Mariana Ionela Mocanu

BUCUREȘTI

2024

UNIVERSITY POLITEHNICA OF BUCHAREST
FACULTY OF AUTOMATIC CONTROL AND COMPUTERS
COMPUTER SCIENCE DEPARTMENT



DIPLOMA PROJECT

Aplicație web pentru managementul învățării

Panait Ștefan-Andrei

Thesis advisor:

Prof. dr. ing. Mariana Ionela Mocanu

BUCHAREST

2024

1 CUPRINS

Sinopsis	3
Mulțumiri	4
1 Introducere	5
1.1 Context	5
1.2 Problema	5
1.3 Obiective	6
1.4 Structura lucrării.....	7
2 Analiza și specificarea cerințelor.....	8
2.1 Cerințe nonfuncționale	8
2.2 Cerințe funcționale.....	9
3 Studiu de piață	12
4 Soluția propusă	20
4.1 Spring Boot	20
4.2 Angular	20
4.3 MySQL	21
4.4 Open-Source.....	21
4.5 Arhitectura bazei de date.....	22
4.6 Docker	23
4.7 Principiul dezvoltare continua si livrare continua.....	23
4.8 Descrierea fluxului funcționalităților	24
5 Detalii de implementare	26
5.1 Autentificare si securitate	27
5.1.1 Autentificare pe baza cookie-urilor	27
5.1.2 OAuth 2.0	27
5.1.3 JWT token	28
5.2 Accesul la baza de date	31
5.2.1 JDBC Template	31
5.2.2 MyBatis	32
5.2.3 Spring Data JPA	32
5.2.4 Implementarea JPA.....	33

5.3	Paginare si căutare după diferite informații	34
5.3.1	Detalii de implementare	36
6	Studiu de caz / Evaluarea rezultatelor	37
6.1	Analiza Web Vitals.....	37
6.2	Analiza Lighthouse	39
6.3	Evaluarea aplicației	40
6.4	Dezvoltări viitoare	48
6.4.1	Migrarea fișierelor către un sistem ca Firebase	48
6.4.2	Crearea unui nou rol de utilizator.....	49
7	Concluzii	49
8	Bibliografie	49

SINOPSIS

Aplicația web pentru managementul învățării este o platformă ușor de utilizat pentru facilitarea educației și a studiului suplimentar, online. Aceasta are o interfață prietenoasă fiind foarte fiabilă atât pentru profesori cât și pentru cursanți în vederea interacționării, gestionării cursurilor. Cursanții înscriși pot să primească recenzii personalizate și să acceseze diferite resurse materiale educative.

Aplicația oferă siguranța utilizatorilor prin autentificare securizată și controlul accesului pe baza de rol pentru modificarea permisiunilor la funcționalități. Printre alte opțiuni, utilizatorul își poate personaliza profilul, atât în rolul de student cât și de profesor. Profesorii pot edita modulele cursurilor și emblema, pot da teme suplimentare și materiale elevilor, precum și recenzii personalizate și note după verificarea temelor efectuate de cursanți.

MULȚUMIRI

Sincere mulțumiri doamnei Prof. dr. ing. Mariana Ionela Mocanu pentru îndrumarea oferită pe parcursul dezvoltării lucrării de licență. Vă mulțumesc pentru toate sfaturile acordate și pentru recenziile și sugestiile constante pe tot parcursul.

1 INTRODUCERE

În domeniul educației, nevoia de inovație este constantă. Atât contextul mondial cât și tehnologia dezvoltată din ultimii ani au atins inevitabil și anii de temelie din viața fiecărui individ, mai exact, anii de școală astfel, modernizarea învățământului reprezintă o prioritate. Cu o tranziție neprevăzută și foarte rapidă în domeniul învățământului, profesorii și elevii au fost și sunt nevoiți să utilizeze diferite platforme online pentru continuarea predării și a studiului. Este evident faptul că nevoia de performanță, flexibilitate, cooperare și accesibilitate nu a fost niciodată la fel de mare.

Aplicația de management al învățării pe care o prezint se adresează acestor nevoi, oferind un mediu online complex pentru facilitarea și sprijinul educației și al dezvoltării. Această platformă este creată pentru îmbunătățirea experienței educaționale a studenților, a profesorilor, dar și a unităților de învățământ.

La bază, aplicația oferă o interfață centralizată, prietenoasă cu utilizatorii și clară, care favorizează comunicarea dintre profesori și cursanți. Cu capacități ce variază de la crearea cursului, gestionarea și editarea acestuia până la livrarea de conținut educativ suplimentar și evaluare detaliată și notarea temelor efectuate și trimise profesorului prin intermediul platformei, aplicația este de un real ajutor în interacțiunea profesor-cursant și facilitarea înțelegerii noțiunilor precum și suplimentarea orelor standard. Focalizându-se pe accesibilitate și performanță la nivel înalt, aplicația web asigură utilizatorilor navigarea cu ușurință pe platformă, accesarea resurselor materiale. Astfel, aplicația asigură elevilor siguranța că educația lor nu este afectată și pot continua călătoria lor de dezvoltare constant, indiferent de factorii care pot părea că împiedică continuitatea educației.

De altfel, consider că fiabilitatea și ușurința folosirii aplicației pot fi cele mai importante atuuri deoarece astfel poate fi la îndemâna oricui, fără a fi nevoie de timp suplimentar special pentru înțelegerea și învățarea folosirii aplicației.

1.1 Context

Situațiile neprevăzute din ultimii ani au dus către o schimbare drastică a organizării învățământului, de la metodele tradiționale de educație, la un nou tip de educație, cel online, care încă tinde să se dezvolte din ce în ce mai mult. Această schimbare a evidențiat importanța și nevoia unei platforme digitale bine dezvoltate, ușor de folosit, în vederea asigurării continuității educației.

1.2 Problema

Învățarea suplimentară poate deveni greu de realizat în contextul tradițional în care profesorii și elevii sunt constrânși și nu își pot desfășura orele și materialele după cum doresc, adaptându-se în funcție de fiecare elev în parte. În continuare, voi enumera o parte din inconveniente ce restricționează învățarea suplimentară în contextul tradițional:

- Limitările de timp
- Limitările de spațiu

- Numărul redus al resurselor materiale puse la dispoziție
- Lipsa interactivității și modernizării
- Lipsa recenziilor detaliate și personalizate
- Lipsa flexibilității
- Lipsa temelor personalizate în funcție de nevoile elevilor
- Constrângerea folosirii cărților impuse de școală

1.3 Obiective

Obiectivul principal pe care mi l-am propus în crearea unei noi platforme de învățare, în scopul învățării suplimentare, îl constituie facilitarea, sprijinirea, inovarea și îmbunătățirea calității învățării suplimentare, atât pentru studenți cât și pentru profesori.

Consider că învățarea suplimentară trebuie să fie la îndemâna oricui și ușor de accesat și folosit, de aceea doresc să evidențiez o parte dintre beneficiile pe care platforma le oferă profesorilor și studenților.

a) Beneficii pentru profesor:

1. Flexibilitate în adăugarea resurselor materiale educative: Profesorii pot încărca oricând materialele educative care le sunt disponibile cursanților în orice moment. De asemenea, profesorii pot adăuga materiale selectate de ei, nu doar din cărțile de la școală.
2. Crearea și gestionarea temelor: Profesorii pot adăuga diferite probleme în lista de exerciții, de asemenea aceștia pot selecta orice problemă din listă pentru a crea o temă pentru elevi. Temele pot fi personalizate pentru nevoile comune ale mai multor studenți sau personalizate pentru un anumit elev. Profesorul poate impune sau nu un anumit termen limită pentru realizarea și încărcarea temelor de către elevi.
3. Evaluarea temelor: După efectuarea și încărcarea temelor de către elevi, profesorii le pot corecta oricând. Corectarea temelor se poate realiza prin citirea lor și oferirea de recenzii personalizate fiecărui elev, cu exemple specifice din tema sa, incluzând și nota.

b) Beneficii pentru studenți:

1. Flexibilitatea în accesarea materialelor educaționale oferite de profesori: Studenții pot vizualiza și descărca resursele materiale oferite de profesori și disponibile în orice moment.
2. Încărcarea temelor: Elevii pot încărca temele efectuate asigurându-se că lucrările lor urmează să fie evaluate de profesori și să primească comentarii clare și precise cu greșelile făcute și sugestiile primite de către cadrele didactice. De asemenea, elevii dispun de flexibilitate în efectuarea și încărcarea pe platformă a temelor rezolvate.

3. Evaluare: După evaluarea temelor, elevii pot citi notițele și sugestiile profesorilor în timp ce își pot revedea lucrarea, cât și rezolvarea corectă a acesteia. De asemenea, fiecare elev primește nota și recenzia în mod privat.

1.4 Structura lucrării

Structura pe care am ales să dezvolt lucrarea este asemănătoare cu structura oferită de facultate.

1. Primul capitol conține o descriere generală a aplicației.
2. Al doilea capitol este împărțit în două subcapitole în care voi prezenta cerințele nonfuncționale și cerințele funcționale propuse.
3. Cel de-al treilea capitol conține realizarea unui studiu de piață.
4. În al patrulea capitol prezint soluția propusă, în special tehnologiile folosite în implementarea aplicației.
5. În cel de-al cincilea capitol ating detaliile de implementare, punând accent în special pe părțile unde a trebuit să iau decizii considerabile și să rezolv probleme.
6. Capitolul șase prezintă evaluarea rezultatelor, precum și dezvoltările propuse pentru viitor.
7. Al șaptelea capitol conține concluziile la care am ajuns în dezvoltarea aplicației.

2 ANALIZA ȘI SPECIFICAREA CERINȚELOR

Putem remarca faptul că în România cultura meditațiilor este una foarte dezvoltată, fiecare părinte își trimite copilul la meditații măcar în cei mai importanți ani din ciclul de învățământ al copiilor acestora. Un studiu la nivel național realizat de IRES și Societatea Academică din România spune că „Conform studiului realizat de IRES și SAR în rândul părinților, o treime dintre elevi participă la sistemul de meditații, cu o frecvență considerabilă în clasele terminale, atunci când examenele naționale se apropie. În urma chestionării elevilor de gimnaziu și liceu, a rezultat faptul că un sfert dintre elevii din România accesează sistemul de meditații la cel puțin o materie.”

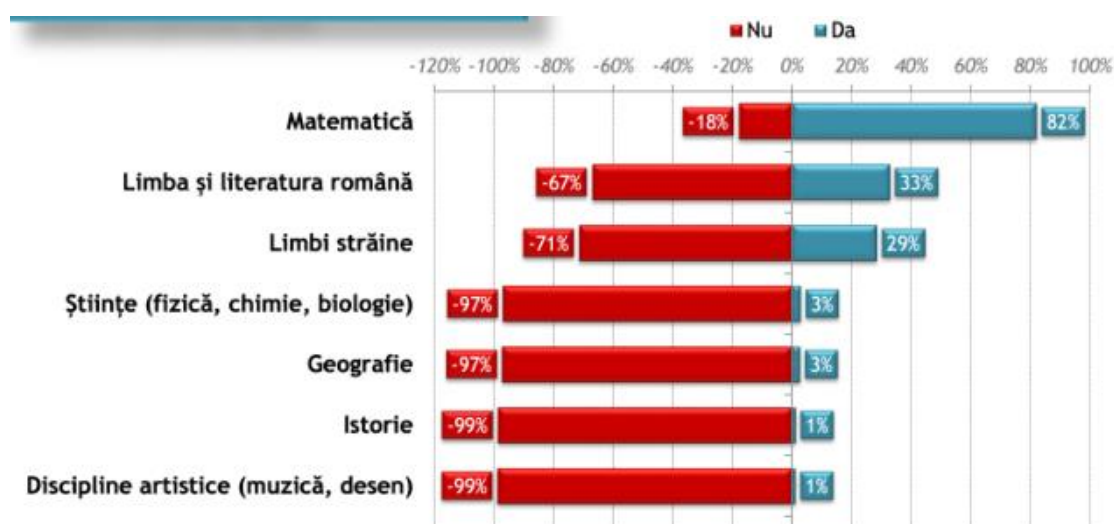


Figura 1: Grafic ce reprezintă la ce materii au elevii nevoie de meditații

Comparând aceste informații cu alte țări:

- În Italia 1 din 5 elevi fac meditații conform unei statistici realizate pe 6000 de elevi de repetizioni.it, platforma italiană pentru meditații atât online, cât și fizic.
- În Germania, la fel ca în Italia, 1 din 5 elevi se adresează profesorilor pentru meditații în învățământul liceal conform lucrării științifice „Effectiveness of private tutoring during secondary schooling in Germany: Do the duration of private tutoring and tutor qualification affect school achievement?”.

Din aceste statistici putem concluziona că studenților nu le sunt de ajuns orele de pregătire din cadrul școlilor. Astfel, această platformă ar acționa ca un pilon educațional pentru studenți.

2.1 Cerințe nonfuncționale

Cerințele nonfuncționale se referă la specificațiile care descriu cum ar trebui să se comporte un sistem, acestea se concentrează pe calitatea și performanța aplicației și sunt esențiale pentru a asigura o experiență optimă pentru utilizatori.

1. Securitate: datele utilizatorilor sunt protejate la orice moment de timp.
2. Performanta: timpii de răspuns și viteza aplicației să nu fie un impediment in folosirea acesteia.
3. Viteza: Aplicația trebuie să răspundă cerințelor într-un timp optim.
4. Fiabilitate: Aplicația va funcționa fără erori care să afecteze disponibilitatea aplicației.

2.2 Cerințe funcționale

Cerințele functionale se axează pe cum aplicația funcționează daca atinge sau nu anumite caracteristici dorite. In aplicația propusă sunt 3 roluri: student, profesor și administrator. In continuare voi prezenta procesul de folosire a aplicației atașând diagrame ale cazurilor de utilizare.

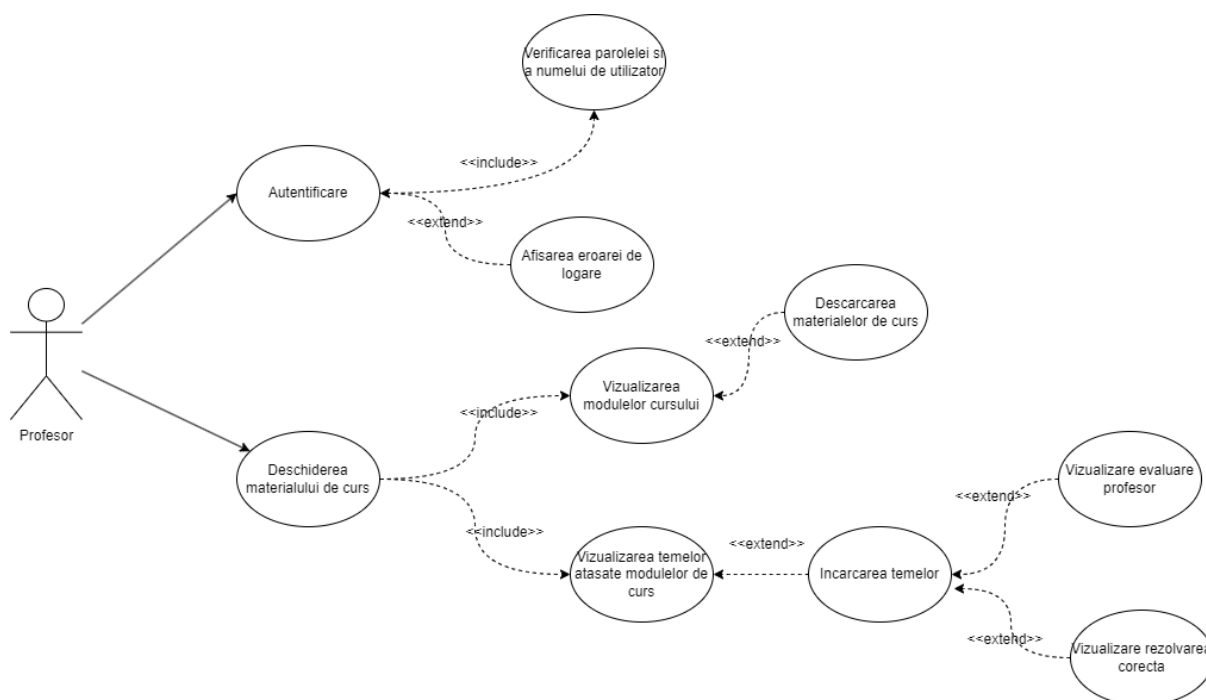


Figura 2: Diagrama de utilizare a unui utilizator în rolul de student

In rolul de student procesul este următorul:

1. Crearea contului: Oricărei creare de cont nou îi este atribuit automat rolul de student.
2. Autentificarea: La autentificare, studentul își adaugă credențialele urmând ca acestea să fie verificate. Dacă acestea sunt corecte, utilizatorul este redirecționat pe pagina unde sunt vizibile cursurile. Dacă credențialele sunt incorecte, acestuia îi va fi afișat un mesaj de eroare.
3. Înscrierea la cursuri: Utilizatorul se poate înscrie singur la cursurile la care crede ca are nevoie de studiu suplimentar.
4. Deschiderea materialului de curs: După înrolarea la curs este accesibilă deschiderea și privirea materialelor disponibile.
5. Vizualizarea modulelor cursului: După deschiderea materialului de curs sunt vizibile modulele cursului. Un exemplu de modul: Titlul: „Polinoame”, Descriere: „Acest

modul te va ajuta să înțelegi polinoamele” și un fișier atașat Suport curs: polinoame.pdf.

6. Descărcarea suportului de curs: Studentul poate descărca suportul de curs.
7. Vizualizarea temelor atașate modulelor de curs: Pentru fiecare modul cursantul poate avea atașat una sau mai multe teme.
8. Încărcarea temelor: După ce studentul consideră că tema este rezolvată acesta poate încărca tema pentru a primi evaluarea unui profesor.
9. Vizualizare evaluare profesor: Elevul poate observa comentariile profesorului asupra temei încărcate.
10. Vizualizare rezolvare corectă: După evaluarea temei de către profesor rezolvarea corectă a exercițiului este disponibilă elevului.



Figura 3: Diagrama de utilizare a unui utilizator în rolul de administrator

În rolul de administrator procesul este următorul:

1. Crearea de utilizator: Administratorul poate crea utilizatori.
2. Modificarea informațiilor utilizatorilor cat și a rolului de student sau profesor: Administratorul poate schimba informațiile utilizatorului de exemplu numele, rolul sau adresa de email.

3. Ștergerea de utilizatori.
4. Crearea de cursuri: Administratorul poate crea cursuri.
5. Modificarea informațiilor cursurilor: Administratorul poate schimba titlul, descrierea cât și categoria din care face parte cursul.
6. Ștergerea de cursuri.
7. Înrolarea de studenți sau profesori la cursuri: Elevul se poate înrola și singur la cursuri, dar profesorul se poate înrola doar dacă este înrolat de administrator. Rațiunea din spatele acestei funcționalități este că doar un administrator poate ști dacă un profesor este bun la o materie. Astfel, nu se va întâmpla, de exemplu, ca un profesor de Informatică să corecteze și să pună materiale la Geografie.
8. Vizualizare listă de probleme: Administratorul poate vizualiza lista de probleme, acesta poate edita problemele sau de asemenea șterge probleme.
9. Adăugare de probleme: Administratorul de asemenea poate adăuga probleme.

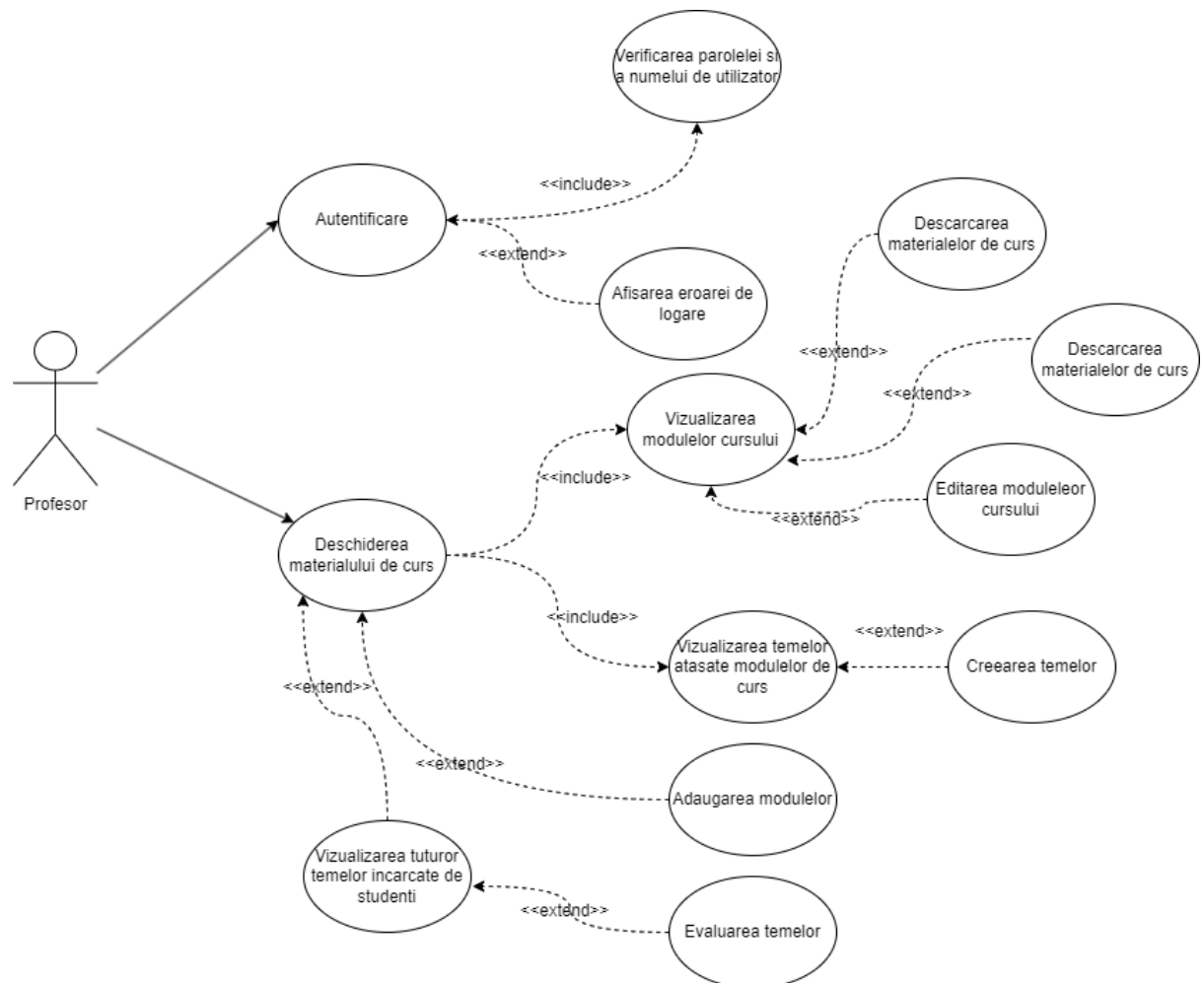


Figura 4: Diagrama de utilizare a unui utilizator în rolul de profesor

În rolul de profesor procesul este următorul:

1. Editarea siglei de curs: Profesorul poate schimba sigla la cursurile la care acesta este înrolat.
2. Adăugarea de probleme în lista de probleme: Profesorul poate adăuga probleme.
3. Adăugarea modulelor cursurilor: Acesta poate adăuga module noi pentru curs.
4. Editarea modulelor de curs: Acesta poate edita titlul modulului, descrierea, cât și poate șterge sau încărca suportul de curs aferent modulului.
5. Vizualizare listă de probleme: Profesorul poate vizualiza lista de probleme pentru a alege o problemă din acea listă pe care să o dea ca temă.
6. Adăugare de probleme: Acesta poate să adauge probleme.
7. Creare de temă: Profesorul poate crea teme alegând o problemă din lista de probleme. Acesta poate, de asemenea, să seteze o zi limită pentru încărcarea acestora.
8. Vizualizarea listei cu elevii care au încărcat temele: Profesorul poate vizualiza lista cu cei care au încărcat temele, apoi cu ajutorul unui modul de căutare, acesta poate căuta după mai multe criterii, cum ar fi titlul temei sau numele elevului.
9. Evaluarea temei: Acesta poate da o recenzie temelor încărcate, cât și o notă.

3 STUDIUL DE PIAȚĂ

Obiectivul apariției unui produs nou pe piață trebuie să se rezume în totalitate la rezolvarea problemelor potențialilor utilizatori, astfel încât fiecare aplicație disponibilă este orientată unui anumit grup țintă.

Aplicația pe care o propun se orientează către elevi și profesori, cu scopul facilitării procesului de învățare și de predare. Consider că, la momentul actual, în special în România, ideea de meditații este puternic răspândită printre elevi, de multe ori numărul cererilor depășind numărul ofertelor.

De altfel, din cauza contextelor globale din ultimii ani, profesorii au fost nevoiți să se orienteze către metode noi de învățare, mai exact platformele de învățare online, considerate poate chiar „nonconformiste” până în acel moment. Pe parcurs, obișnuindu-se cu noile metode, din ce în ce mai mulți profesori au început să utilizeze platformele online de învățare, considerându-le benefice.

În domeniul educației este nevoie tot timpul de metode noi și inovative care să faciliteze învățarea. Astfel, pentru a realiza o platformă online de învățare suplimentară, utilă și fiabilă, am creat un chestionar orientat către nevoile întâmpinate de elevi, aflând de asemenea și părerea unui număr de 65 de persoane despre ideea de „platformă online de învățare”.

Ce vârstă aveți?

65 responses

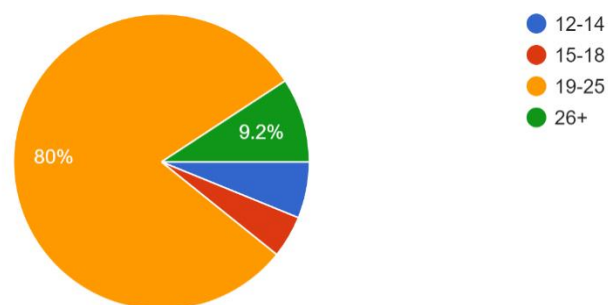


Figura 5

Mediul de proveniență

65 responses

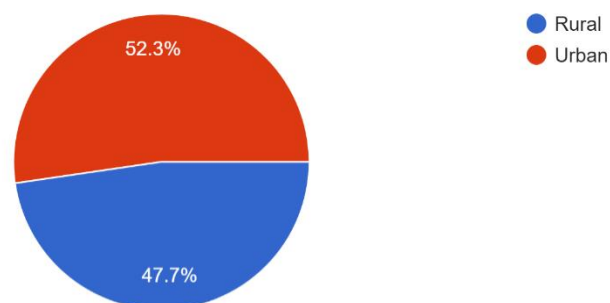


Figura 6

Cu ajutorul figurilor 1 și 2 putem observa fiabilitatea platformelor online de învățare, relevante și benefice persoanelor de diferite vârste, fiind chestionate 65 de persoane cu vârste cuprinse între 12 și peste 26 de ani, provenite atât din mediul urban (52.3%) cât și din cel rural (47.7%).

Ultima formă de învățământ finalizată
65 responses

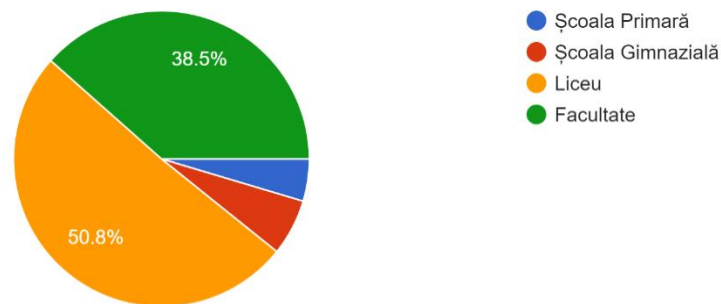


Figura 7

În perioada în care ați fost student/elev, ați simțit nevoia de ore suplimentare, în plus față de cele de la școală?
65 responses

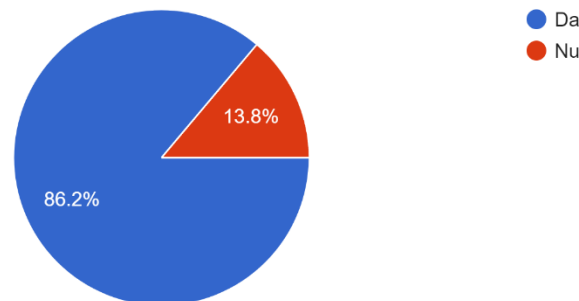


Figura 8

De altfel, cu ajutorul figurilor 3 și 4 observăm faptul că 86,2% dintre persoanele chestionate au simțit nevoia de ore suplimentare în perioada studiilor și doar 13,8% nu au simțit nevoia, persoanele întrebate având ca ultimă formă de învățământ finalizată școala primară, școala gimnazială, liceu și facultate. Astfel, putem spune că nevoia orelor suplimentare apare indiferent de forma de învățământ, indiferent dacă urmează un examen important precum bacalaureatul sau nu.

Este evidențiată nevoia unei schimbări în rândul elevilor și al studenților, pentru facilitarea procesului de învățare al acestora și ajutorarea profesorilor cărora le este greu ca într-un curs să ajungă la 100% numărul elevilor care au înțeles și nu mai au nevoie de pregătire și explicații în plus.

Dacă da, la ce materie ați întâmpinat dificultăți?

65 responses

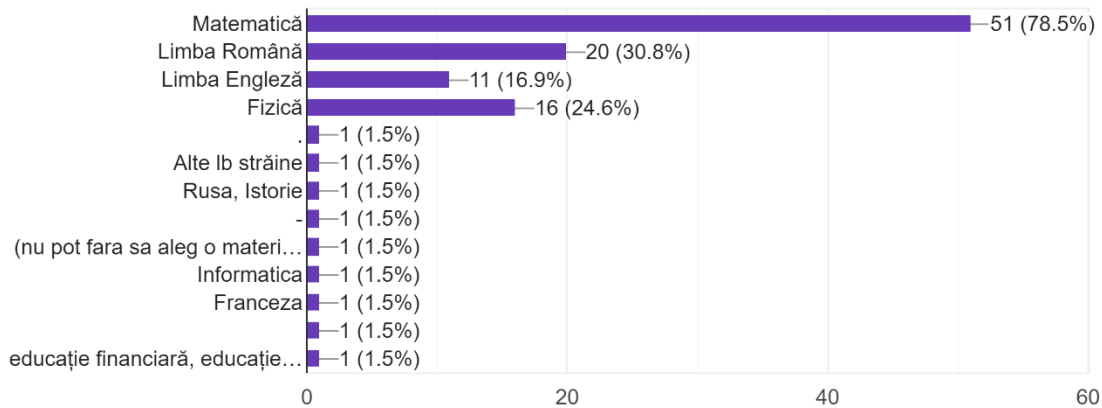


Figura 9

Ați apelat la meditații pe parcursul anilor de învățământ din cauza nevoii de suplimentare a orele de la clasă?

65 responses

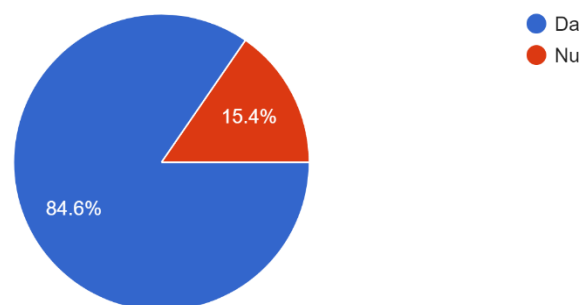


Figura 10

Din figura 9 reiese faptul că doar 4 persoane din 65 nu au întâmpinat dificultăți în perioada școlii. Consider că numărul persoanelor ce nu au avut probleme pe parcursul anilor de studii este unul extraordinar de mic și trebuie schimbat. Acest fapt m-a motivat în realizarea aplicației de învățare suplimentară și sper să reușesc să aduc o îmbunătățire și un beneficiu atât elevilor, cât și profesorilor.

De asemenea, privind figura 10, observăm că 84,6% dintre persoanele chestionate au apelat la meditații din cauza nevoii de a suplimenta orele de la clasă și doar 15,4% nu au făcut meditații. Procentul persoanelor care au făcut meditații este unul foarte mare, de altfel și nevoia de pregătire suplimentară este uriașă. Susțin faptul că platforma de învățare suplimentară poate sprijini nevoile elevilor.

Sunteți de acord că meditațiile NU sunt la îndemâna oricui?

65 responses

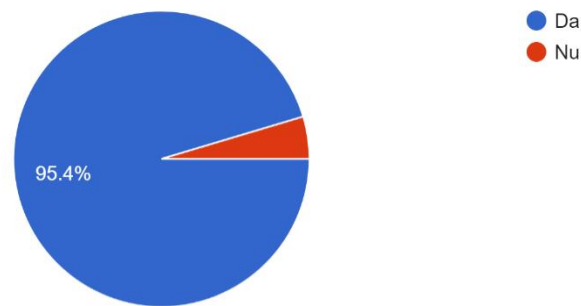


Figura 11

Figura 11 reprezintă un adevăr foarte trist cu care, din păcate, multe persoane se confruntă. Meditațiile, din nefericire, nu sunt la îndemâna oricui, de altfel mulți elevi, chiar dacă au nevoie de pregătire suplimentară, nu pot avea parte de ea, iar acest lucru îi poate împiedica în dezvoltarea lor.

Sunt de părere că toată lumea ar trebui să aibă acces la pregătire suplimentară în caz de nevoie și consider că platforma prezentată poate ajuta prin scăderea numărului elevilor ce nu au acces la pregătire suplimentară și a celor care apelează la meditații.

Considerați că platformele de învățare online (ex Google Classroom, Moodle etc.) sunt benefice atât pentru profesori cât și pentru elevi/studenti?

65 responses

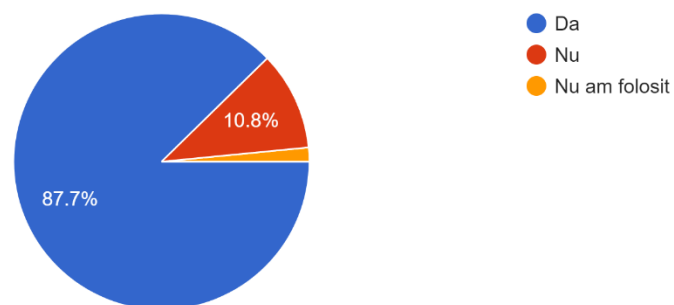


Figura 12

Considerați că o platformă online de învățare suplimentară, nouă și ușor de folosit, poate reduce numărul elevilor și al studenților care apelează la meditații ?

65 responses

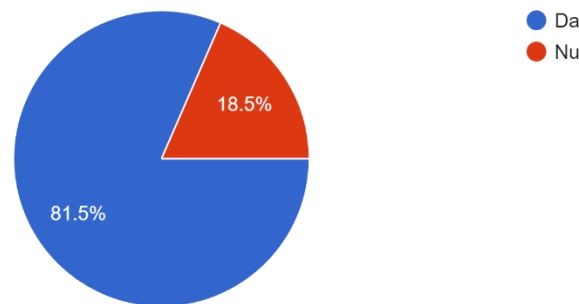


Figura 13

Analizând figurile 12 și 13, vedem părerea persoanelor chestionate referitor la platformele online și beneficiile lor. În figura 8 se observă că doar 10,8% nu consideră benefice aplicațiile de învățare online deja existente pe piață și 1,5% nu au folosit, rămânând un procentaj foarte mare de persoane ce consideră benefice aplicațiile de educație online.

De asemenea, în figura 9 este precizat faptul că 81,5% dintre persoane sunt de părere că o nouă platformă online de învățare suplimentară poate reduce numărul elevilor care apelează la meditații, întărind ideea pe care o susțin și obiectivul pe care mi l-am propus în realizarea aplicației prezentate.

Care sunt motivele pentru care ați folosi o platformă online de învățare suplimentară?

65 responses

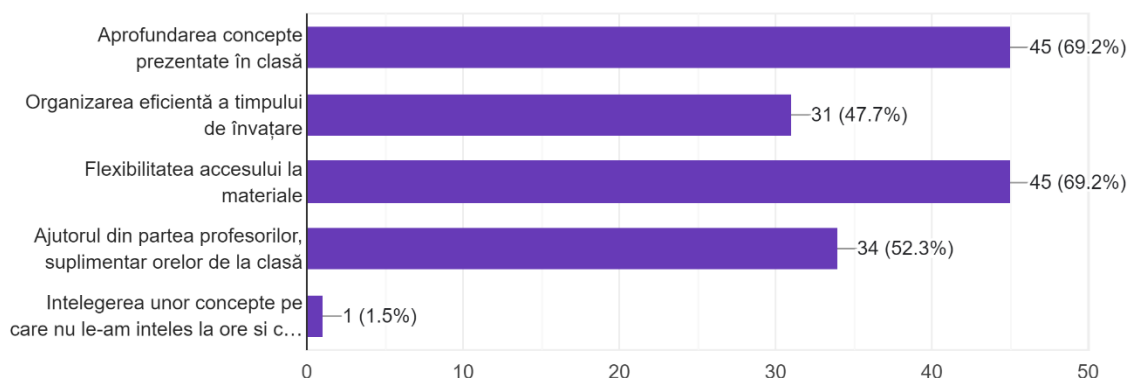


Figura 14

Care considerați că ar fi beneficiile combinării platformei online de învățare suplimentară cu metoda tradițională (orele de la școală)?

65 responses

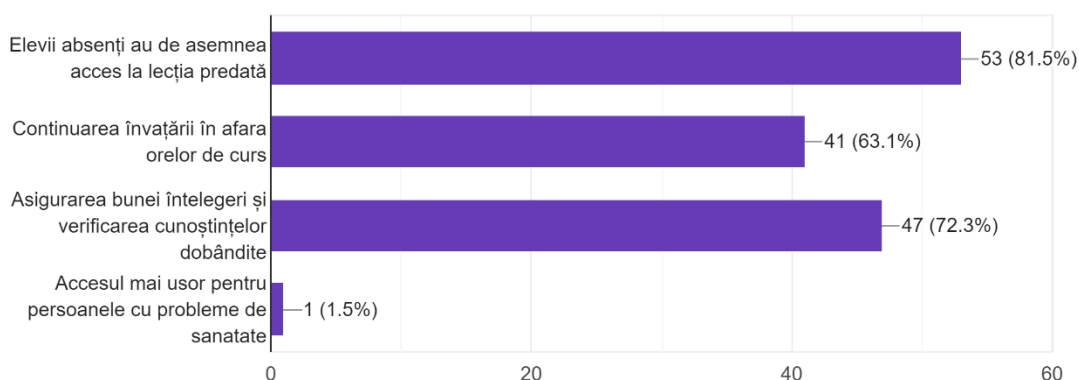


Figura 15

Care considerați că ar fi beneficiile studenților folosirii platformei online de învățare suplimentară?

65 responses

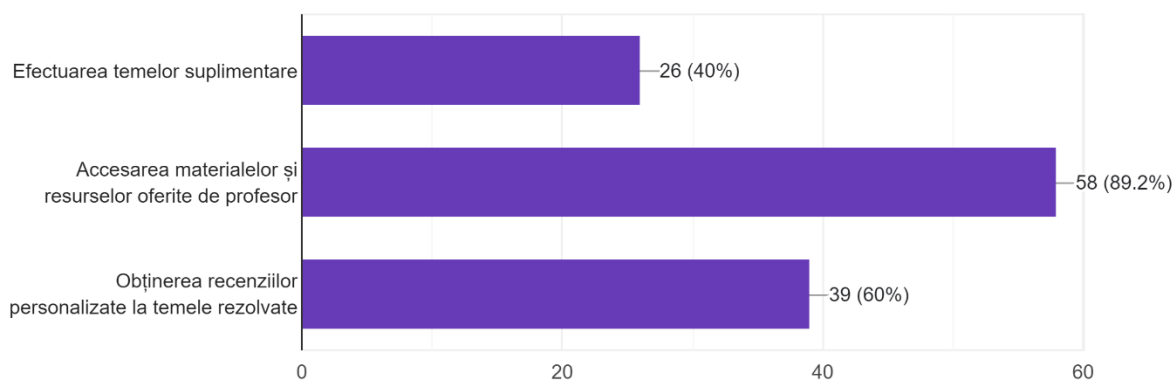


Figura 16

În prezent, în multe școli și instituții de învățământ, profesorii folosesc platformele de învățare online pentru trimiterea temelor sau, mai rar, pentru postarea cursurilor din ziua respectivă. Însă, platforma pe care eu o propun vizează în special învățarea suplimentară și înlocuirea pe cât posibil a meditațiilor prin activități, cursuri și explicații suplimentare, și nu strict pentru trimiterea temelor sau a lecției de zi.

Consider că o platformă online nouă și ușor de folosit poate fi benefică învățământului și dezvoltării persoanelor.

Observăm în figura 10 că niciuna dintre persoanele chestionate nu a menționat faptul că nu ar dori să folosească o asemenea platformă, în plus răspunsurile primite sunt pro unei asemenea platforme deoarece nevoia învățării suplimentare este mare.

De asemenea, figurile 11 și 12 se referă la beneficiile platformei, în perspectiva elevilor dar și a profesorilor. În plus, niciuna dintre persoanele chestionate nu a menționat vreun dezavantaj în folosirea acestei noi platforme.

Pentru a exemplifica, analizăm figura 11 care se referă la imaginea de ansamblu și beneficiile pe care platforma le poate avea în învățământ, prin combinarea sa cu metoda de învățare tradițională, cea din sălile de clasă. Persoanele chestionate consideră un avantaj combinarea celor două metode de învățare, în vederea unei mai bune înțelegeri a cursurilor, continuarea învățării și comunicarea cu profesorii în afara programului, în plus, chiar sugestia anonimă privind accesul persoanelor cu probleme de sănătate oferă o imagine de ansamblu asupra câștigului utilizării platformei. Se observă faptul că verificarea cunoștințelor dobândite are o mare importanță și este pe locul 2 în clasament, la mică diferență de locul 1 care vizează accesul elevilor absenți la materialele și cursurile predate în ziua respectivă.

Ultima întrebare a chestionarului (figura 12) îi vizează strict pe elevi și avantajele pe care aceștia le-ar avea utilizând platforma prezentată. De asemenea, fiecare persoană a ales drept răspuns minim două variante, rezultând beneficiile și suportul pe care cursanții le-ar avea prin folosirea platformei online de învățare suplimentară.

Consider că răspunsurile primite în acest chestionar îmi întăresc și îmi sprijină ideea de necesitate a unei asemenea platforme, ce poate fi la îndemâna cât mai multor persoane și poate aduce ajutor învățământului actual și elevilor ce se confruntă cu diferite probleme privind educația, prin fiabilitatea sa, ușurința folosirii din orice tip de profil și opțiunile pe care aceasta le oferă profesorilor și elevilor.

4 SOLUȚIA PROPUȘĂ

Abordarea pe care am ales să merg provine din documentarea pieței la momentul actual. Astfel, după o analiză amănunțită, am observat că o bună parte din marile companii precum JPMorgan Chase Bank, Intellibus, Deutsche Bank și multe altele folosesc Java Spring Boot pentru backend. Pentru frontend am ales să merg pe Angular folosind TypeScript, cu toate că React este mai popular în rândul aplicațiilor mai mici, dar pentru proiectele mari este mai folosit Angular. Pentru baza de date am ales să merg pe o bază de date relațională, respectiv MySQL, aceasta fiind open-source, de asemenea fiind cunoscută pentru performanță și scalabilitate.

4.1 Spring Boot

Spring este cel mai popular framework de Java. Titulatura de cel mai popular framework este obținută din viteza, simplitatea și productivitatea acestuia.

Spring este un framework pentru a construi aplicații Java, open-source, care oferă o infrastructură complexă și ușor de menținut pentru dezvoltarea aplicațiilor. Principalele avantaje în folosirea Spring:

1. Inversion of control – acesta este un principiu de proiectare care oferă modularitate programului, oferind frameworkului control asupra unor porțiuni de cod sau a unor obiecte. Acest lucru este realizabil prin Dependency Injection. Dependency Injection este mecanismul prin care controlul este inversat de la programator la framework.
2. Accesul la date – simplificarea accesului la date folosind tehnologii precum Hibernate, JPA (Java Persistence API), MyBatis, EclipseLink. Acestea reduc complexitatea interacțiunii cu baza de date.
3. Ideal pentru construirea de RESTful API datorită adnotărilor simpliste: @RestController, @GetMapping, @PostMapping simplifică definiția și gestionarea. Aceste adnotări fac codul ușor de citit și întreținut, permițând o dezvoltare facilă a endpointurilor de tip CRUD (create, read, update, delete).
4. Securitatea – securitatea este ușor de facilitat datorită frameworkului Spring Security JWT (JSON-Web Token) care ajută în autentificare și autorizare. De asemenea, oferă protecție împotriva atacurilor de tip CSRF.

4.2 Angular

Angular este un framework open-source bazat pe TypeScript, care rulează pe Node.js și este utilizat pentru crearea aplicațiilor web.

De ce TypeScript și nu JavaScript?

Unul dintre cele mai mari avantaje pe care consider că le are TypeScript asupra JavaScript este că acesta are erori de compilare datorită tipizării statice, în loc de tipizarea dinamică în JavaScript.

Caracteristici ale Angular

Angular este bazat pe componente. Componentele oferă utilizatorului abilitatea de a defini propriile componente personalizate. Acestea pot fi folosite simultan cu elementele HTML standard, precum `<div></div>`. Un exemplu de folosire al acestei caracteristici este crearea unei componente `<add-problem></add-problem>` care este o pagină cu un formular pentru adăugarea exercițiilor. Aceasta poate fi folosită ca un „obiect” în orice altă componentă a aplicației. Astfel, acest stil de programare este similar cu programarea orientată pe obiecte, aducând toate beneficiile acesteia, cum ar fi modularitatea, reutilizarea, întreținerea și posibilitatea de extindere.

Printre beneficiile majore ale utilizării componentelor se numără:

1. Organizare și Claritate: Codul este mult mai bine organizat și clar, facilitând înțelegerea și navigarea.
2. Reutilizarea Codului: Componentele pot fi reutilizate în diferite părți ale aplicației, reducând redundanța și economisind timp.
3. Întreținere Facilă: Codul modular și componentizat este mai ușor de întreținut și actualizat, permițând implementarea mai rapidă a modificărilor și a noilor funcționalități.

4.3 MySQL

MySQL este un sistem de gestionare a bazelor de date relaționale (RDBMS) open-source. Este unul dintre cele mai populare și utilizate motoare de baze de date din lume, cunoscut pentru performanța sa, fiabilitatea și ușurința în utilizare. MySQL este dezvoltat, distribuit și susținut de Oracle.

4.4 Open-Source

Am menționat faptul că alegerile mele în construirea aplicației sunt open-source. Astfel, voi argumenta care sunt avantajele:

1. Transparența: Dezvoltatorii pot inspecta codul sursă pentru a observa dacă acesta este funcțional. De asemenea, acesta este un factor bun și din punct de vedere al securității.
2. Securitatea: Datorită transparenței, dezvoltatorii pot analiza codul de vulnerabilități ascunse. Legea lui Linus Torvalds „Given enough eyeballs, all bugs are shallow” – aceasta înseamnă că, cu cât mai multă lume lucrează și testează aplicația, problemele aplicației dispar.
3. Valoare educațională.
4. Suport și comunitate: Datorită faptului că este open-source, suportul și comunitatea sunt pe măsură, fiind foarte ușor să găsești informații. De asemenea, toți dezvoltatorii pot aduce contribuții.
5. Inovare și dezvoltare rapidă: Datorită faptului că toată lumea poate contribui, se ajunge la dezvoltarea rapidă și aducerea de noi inovații în tehnologie.

4.5 Arhitectura bazei de date

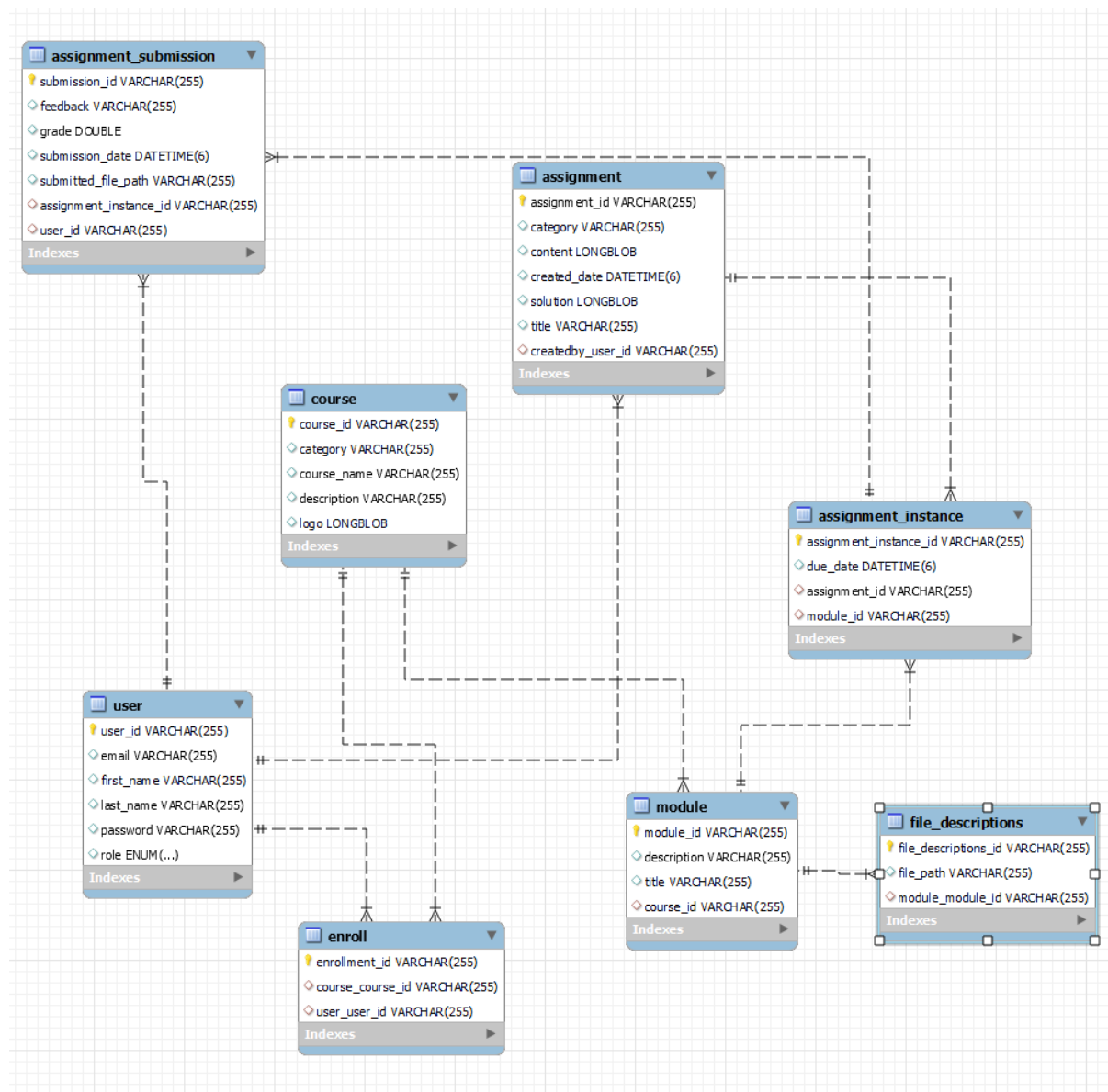


Figura 17: Structura bazei de date

1. Tabela user: Aceasta conține informațiile despre utilizatori. Coloanele sunt: user_id (cheie primară), email, first_name, last_name, password și role.
2. Tabela course: Aceasta conține informațiile despre cursuri. Coloanele sunt: course_id (cheie primară), category, course_name, description și logo.
3. Tabela module: Aceasta conține informații despre modulele din curs, fiecare modul fiind legat de un curs. Coloanele sunt: module_id (cheie primară), description, title și course_id (cheie străină).
4. Tabela file_description: Aceasta conține căile pentru fișierele din module. Coloanele sunt: file_description_id (cheie primară), file_path și module_module_id (cheie străină).

5. Tabela assignment: Aceasta conține informații despre exercițiile care pot fi folosite ca teme. Coloanele sunt: assignment_id (cheie primară), category, content, created_date, solution, title și created_by_user_id (cheie străină).
6. Tabela assignment_instance: Aceasta tabelă conține informații despre teme, fiind legată de module. Coloanele sunt: assignment_instance_id (cheie primară), due_date, assignment_id (cheie străină) și module_id (cheie străină).
7. Tabela assignment_submission: Aceasta tabelă conține informații despre temele încărcate de studenți. Coloanele sunt: submission_id (cheie primară), feedback, grade, submission_date, submitted_file_path, assignment_instance_id (cheie străină) și user_id (cheie străină).
8. Tabela enroll: În aceasta tabelă este reținut ce utilizator este înregistrat la un anumit curs. Coloanele sunt: enrollment_id (cheie primară), course_course_id (cheie străină) și user_user_id (cheie străină).

4.6 Docker

Docker este o platformă open-source care automatizează deployment-ul aplicațiilor, permițând rularea acestora într-un mediu izolat. Docker facilitează împachetarea unei aplicații într-un singur container, ceea ce contribuie la principiile de dezvoltare continuă și livrare continuă.

În aplicație vor fi trei servicii: frontend, backend și baza de date, orchestrate printr-un fișier docker-compose.yml. Fiecare serviciu rulează într-un container propriu și este interconectat printr-o rețea comună. Astfel, vor rula trei containere:

1. mysqldb: Containerul care rulează MySQL.
2. Spring Boot RESTful Web Services: Containerul care rulează backend-ul aplicației, configurat pentru a se conecta la baza de date și la frontend.
3. frontend: Containerul care rulează serverul Angular pentru partea de frontend.

De asemenea, utilizez o rețea comună, springboot-mysql-net, pentru a permite containerelor să comunice eficient între ele.

4.7 Principiul dezvoltare continua si livrare continua

Obiectivul principiului dezvoltare continuă și livrare continuă este de a crește rapiditatea cu care este dezvoltată o aplicație și de a îmbunătăți calitatea produsului finit. Principalul atu al acestui principiu este faptul că îmbunătățește experiența dezvoltatorului, dar și a consumatorului aplicației, aplicația fiind livrată către acesta în cicluri stabilite de timp, permițând o mai bună comunicare între aceștia. Consumatorul poate propune schimbări în aplicație în orice moment de timp, iar dezvoltatorul poate implementa această schimbare și o poate livra într-o nouă iterație.

Astfel, acest principiu se pliază cel mai bine pe un mod de dezvoltare iterativ.

4.8 Descrierea fluxului funcționalităților

Am ales să folosesc o diagramă simplă pentru a evidenția punctele critice de decizie și pentru a ilustra fluxul funcționalităților. Aceste diagrame sunt vitale pentru înțelegerea modului în care diferitele părți ale sistemului lucrează împreună pentru a îndeplini anumite funcționalități.

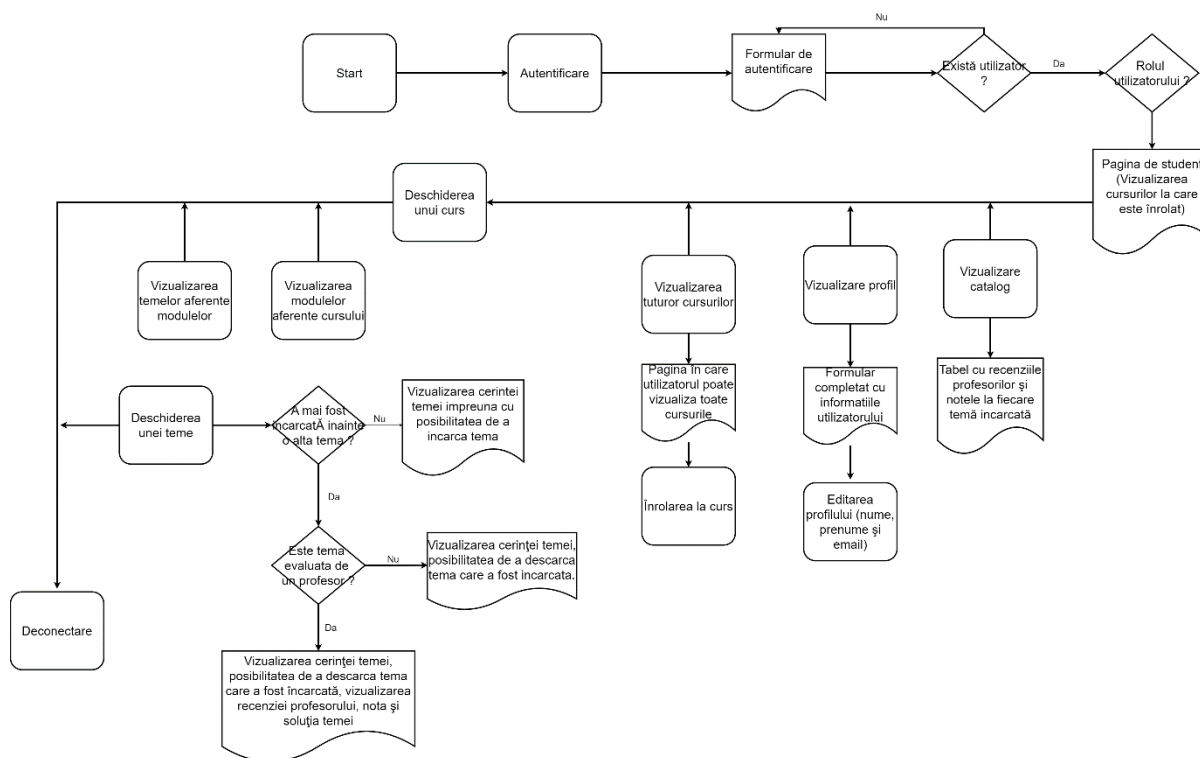


Figura 18: Diagrama de flux pentru utilizatorii cu rolul de student

În această diagramă sunt evidențiate toate funcționalitățile pe care le poate efectua un utilizator în rolul de student. La autentificare, dacă credențialele sunt corecte, acesta este redirectionat pe o pagină pe care poate vizualiza toate cursurile la care este înrolat. Descrierea fluxului funcționalităților din pagina inițială:

- Vizualizarea catalogului cu posibilitatea de a căuta recenziile profesorilor la un anumit curs.
- Vizualizarea profilului cu posibilitatea de a modifica elemente ale profilului, cum ar fi nume, prenume și email.
- Vizualizarea tuturor cursurilor cu posibilitatea de înrolare la orice curs care apare.

Descrierea fluxului funcționalităților după deschiderea unui curs la care studentul este înrolat:

- Vizualizarea modulelor de curs.
- Vizualizarea temelor.

La deschiderea unei teme:

- Dacă o temă nu este încărcată, acesta vede cerința și are posibilitatea de a încărca.

- Dacă este deja o temă încărcată care încă nu este evaluată de un profesor, studentul poate vedea cerința temei, ultima încărcare, posibilitatea de a descărca ultima încărcare și posibilitatea de a încărca pentru a modifica ultima încărcare.
- Dacă tema este evaluată de un profesor, studentul poate observa recenzia profesorului, nota, cât și posibilitatea de a vizualiza cerința, încărca o altă temă și descărca ultima încărcare.

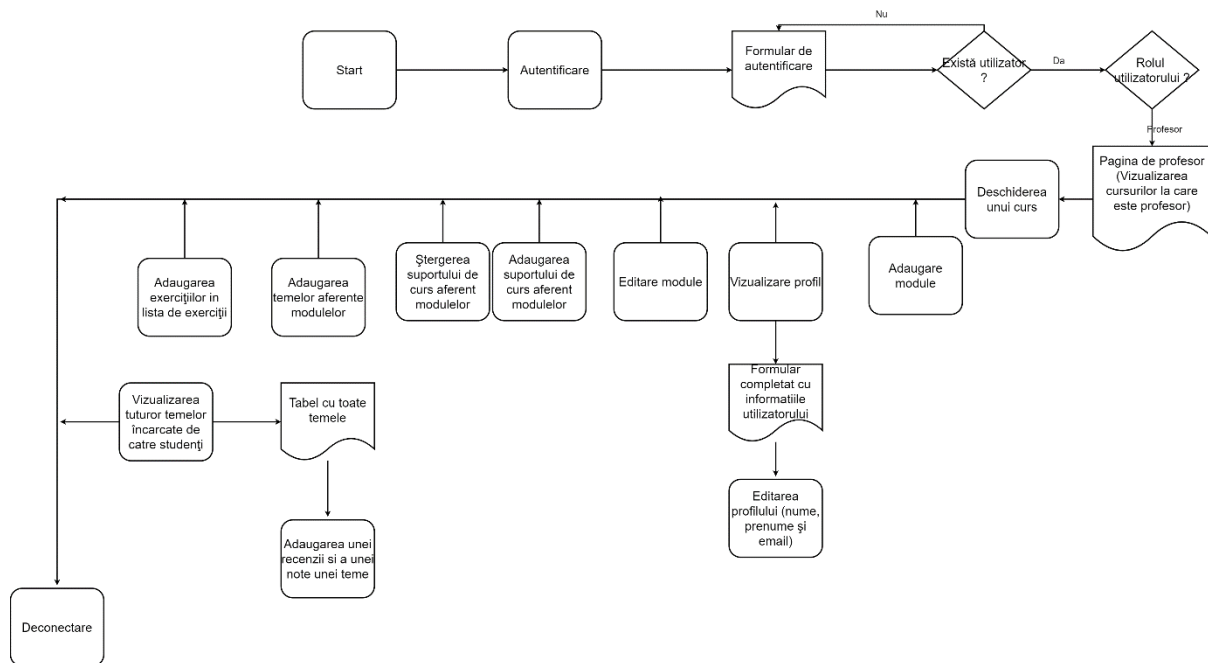


Figura 19: Diagrama de flux pentru utilizatorii cu rolul de profesor

În această diagramă sunt evidențiate toate funcționalitățile pe care le poate efectua un utilizator în rolul de profesor. Dacă profesorul deschide un curs, poate realiza următoarele acțiuni:

- Adăugarea unui modul.
- Editarea unui modul.
- Ștergerea unui modul.
- Adăugarea suportului de curs aferent modulelor.
- Ștergerea suportului de curs aferent modulelor.
- Adăugarea temelor aferente modulelor.
- Adăugarea exercițiilor în lista de exerciții.
- Vizualizarea listei de exerciții, modificarea și ștergerea exercițiilor din listă, la care pentru o mai ușoară folosire există un sistem de căutare specializat.
- Vizualizarea tuturor temelor încărcate de către studenți, unde apare un tabel cu toate temele încărcate pentru cursul respectiv, la care profesorul poate căuta dacă dorește să corecteze temele cu un anumit titlu.

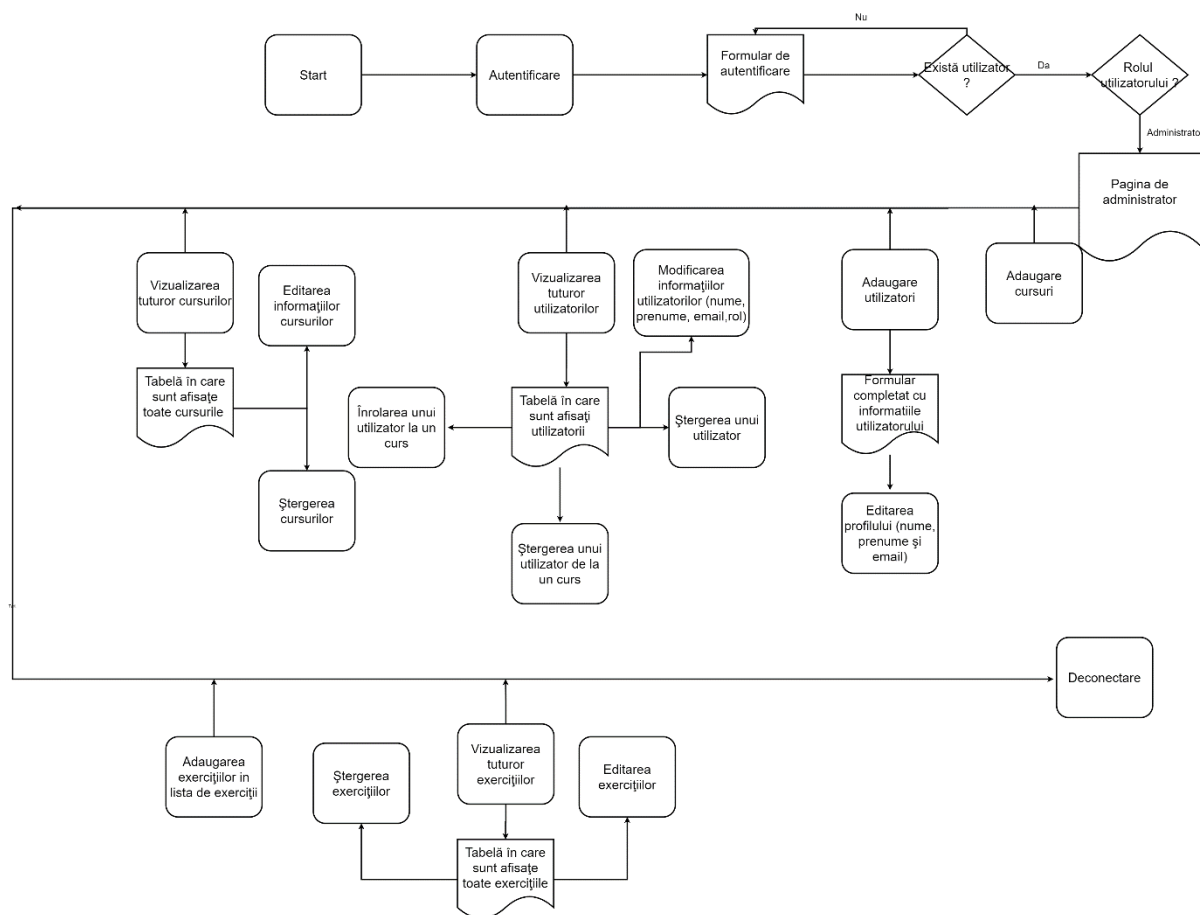


Figura 20: Diagrama de flux pentru utilizatorii cu rolul de administrator

În această diagramă sunt evidențiate toate funcționalitățile pe care le poate efectua un utilizator în rolul de administrator:

- Adăugarea cursurilor.
- Adăugarea utilizatorilor.
- Vizualizarea tuturor utilizatorilor, de unde administratorul poate edita informațiile utilizatorului (nume, prenume, email și rol), înrola profesori și utilizatori la anumite cursuri, șterge utilizatori de la anumite cursuri și șterge utilizatori.
- Vizualizarea tuturor cursurilor, de unde administratorul poate edita informațiile cursului și șterge cursul..
- Vizualizarea listei de exerciții, modificarea și ștergerea exercițiilor din listă, la care pentru o mai ușoară folosire există un sistem de căutare specializat
- Adăugarea exercițiilor în lista de exerciții.

5 DETALII DE IMPLEMENTARE

În acest capitol vom explora în detaliu, intrând amănunțit în provocările întâmpinate pe parcursul dezvoltării aplicației. Pentru a intra mai în amănunt și a oferi o perspectivă clară asupra problemelor, mă voi folosi de prezentarea secvențelor de cod esențiale, pseudo-cod, diagrame de secvență și de resursele găsite în timpul cercetărilor. Resursele prezintă diferite

perspective, incluzând articole, studii de caz și documentații tehnice folosite pentru a privi din perspective diferite soluțiile problemei.

În plus, capitolul va detalia modul în care studiul de caz și tehnologiile propuse au fost de ajutor în dezvoltarea soluției finale, accentuând modul în care ele au contribuit la rezolvarea problemelor. De asemenea, vom analiza strategiile de arhitectură abordate la diferite impedimente în structura aplicației.

Prin prezentarea detaliată a algoritmilor și structurilor utilizate, împreună cu etapele de dezvoltare și corectitudinea soluțiilor implementate, acest capitol va oferi o imagine completă și transparentă a procesului de dezvoltare tehnică.

5.1 Autentificare si securitate

Pentru a înțelege metoda pe care am ales-o în autentificarea și autorizarea utilizatorilor în aplicație, cât și elemente ce țin de securitate, voi prezenta mai multe opțiuni.

5.1.1 Autentificare pe baza cookie-urilor

Autentificarea pe bază de cookie-uri funcționează în următoarea manieră:

1. Utilizatorul completează email-ul și parola și trimite o cerere către server de tip POST.
2. Serverul verifică validitatea credențialelor. Dacă acestea sunt corecte, creează o sesiune pe care o reține și trimite ID-ul sesiunii alături de informațiile aferente într-un cookie către client.
3. La fiecare cerere ulterioară, clientul trimite automat informațiile din cookie. Astfel, serverul poate verifica dacă utilizatorul este autorizat sau nu.

Avantajele Autentificării pe bază de Cookie-uri

- Gestionare Automată: Cookie-urile sunt administrate automat de browser, simplificând astfel codul pe partea de client.
- Validare și Invalidare Facilă: Deoarece serverul menține starea, acesta poate valida sau invalida sesiunile cu ușurință.

Dezavantajele Autentificării pe bază de Cookie-uri

- Scalabilitate: Serverul trebuie să păstreze starea sesiunilor, ceea ce face dificilă scalarea verticală.
- Blocaje la trafic mare: Administrarea sesiunilor pe server poate duce la blocaje în cazul unui trafic mare de cereri.

5.1.2 OAuth 2.0

OAuth 2.0 este un protocol de autorizare care permite aplicațiilor terțe să obțină acces limitat la resursele utilizatorilor fără a necesita trimiterea de credențiale.

Pașii prin care se realizează autorizarea sunt următorii:

1. Clientul este redirecționat către o aplicație terță aleasă, unde utilizatorul introduce credențialele, dacă nu este deja autorizat pe acea aplicație.
2. Aplicația terță autorizează autentificarea și utilizatorul este redirecționat către aplicația principală împreună cu un cod de autorizare.
3. Clientul creează o cerere de tip POST ce conține codul de autorizare de la aplicația terță pentru a obține un token de acces în aplicație.
4. La fiecare cerere ulterioară, tokenul de acces este trimis în antetul cererii, astfel se face autorizarea.

Avantajele autentificării pe bază de OAuth 2.0

- Autentificarea nu este strâns legată de credențialele utilizatorului.
- Suportă diverse tipuri de fluxuri de autorizare și de tokeni, cum ar fi opaque tokens, SAML assertions, JWT tokens.

Dezavantajele autentificării pe bază de OAuth 2.0

- Cea mai mare vulnerabilitate pe care o poate avea acest tip de autorizare este ca codul de autorizare din pasul 2 să fie interceptat de un atacator.
- Se bazează pe o aplicație externă pentru a gestiona autorizarea.

5.1.3 JWT token

Autentificarea pe bază de JSON Web Token este o metodă fără stare ce permite transmiterea de informații între server și client într-un mod securizat prin intermediul unui token semnat. Pentru a descrie pașii de autentificare și autorizare folosind JSON Web Token, voi utiliza o diagramă de secvență (figura 21).

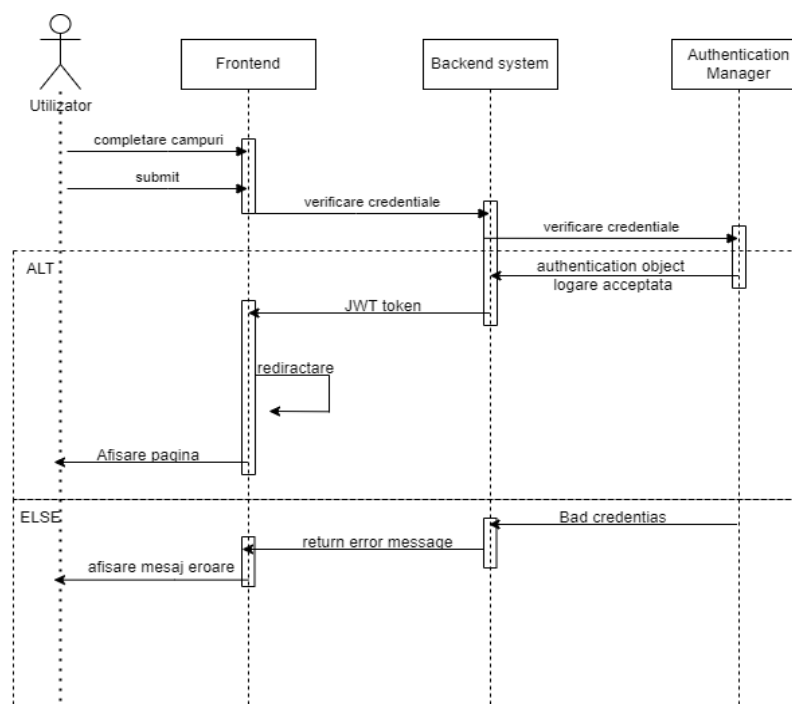


Figura 21: Diagrama de secvență pentru autentificare

Astfel, utilizatorul completează câmpurile (email și parolă) și apasă autentificare. Clientul trimite o cerere de tip POST ce conține credențialele utilizatorului către server. Serverul utilizează un Authentication Manager care facilitează verificarea credențialelor în baza de date. Dacă credențialele sunt corecte, acesta returnează un obiect de autentificare, iar clientul primește un JWT token criptat ce conține proprietățile utilizatorului (email, rol, data expirării tokenului). Clientul îl salvează în memoria locală a motorului de căutare și îl trimite ulterior cu fiecare cerere pe care clientul o face. Dacă credențialele sunt incorecte, Authentication Managerul întoarce HTTP STATUS BAD CREDENTIALS, iar utilizatorului îi este afișat un mesaj de eroare.

Avantajele autentificării pe bază de JSON Web Token (JWT)

- Scalabilitate: Datorită faptului că este stateless, nu necesită menținerea informațiilor de către server, ceea ce face aplicația scalabilă.
- Performanță: Tokenul poate conține informații (claims), eliminând astfel necesitatea unor interogări frecvente în baza de date, cum este cazul autentificării folosind sesiuni cookie.
- Interoperabilitate: Poate fi utilizat cu orice limbaj de programare, iar datorită faptului că este stateless se pliază foarte bine pe principiile RESTful.
- Securitate: Tokenurile sunt semnate digital, asigurând integritatea datelor.

Dezavantajele autentificării pe bază de JSON Web Token (JWT)

- Revocarea tokenului înainte de expirare este complicată. Soluțiile includ utilizarea unei liste de revocare sau emiterea de tokenuri cu durată scurtă de viață.
- Securitate: Deoarece payload-ul (claims) JWT nu este criptat, ci doar semnat, informațiile sensibile nu ar trebui incluse în token.

Rezultatul analizei mele a fost că utilizarea JSON Web Token este cea mai bună alegere pentru această aplicație.

```
no usages
@Bean
public SecurityFilterChain securityFilterChain(HttpSecurity httpSecurity) throws Exception {
    httpSecurity
        .csrf(AbstractHttpConfigurer::disable)
        .cors(httpSecurityCorsConfigurer -> httpSecurityCorsConfigurer.configurationSource(corsConfigurationSource()))
        .authorizeHttpRequests(auth -> auth
            .requestMatchers(...patterns: "/api/auth/**").permitAll()
            .requestMatchers(...patterns: "/api/admin/**").hasRole("ADMIN")
            .anyRequest().authenticated())
        .sessionManagement(sess -> sess.sessionCreationPolicy(SessionCreationPolicy.STATELESS))
        .authenticationProvider(authenticationProvider)
        .addFilterBefore(jwtAuthFilter, UsernamePasswordAuthenticationFilter.class);

    return httpSecurity.build();
}
1 usage
```

Figura 22: Configurarea securității

Principala configurație a securității se află în imaginea din figura 22. Astfel, metoda `securityFilterChain` configurează CORS, definește regulile de autorizare și adaugă filtrul de autentificare JWT înainte de `UsernamePasswordAuthenticationFilter`. Acest filtru este o clasă `JwtAuthenticationFilter` care extinde `OncePerRequestFilter`, asigurând faptul că filtrul este apelat o singură dată pe cerere și suprascrie metoda `doFilterInternal` care verifică antetul „Authorization” pentru a extrage și valida tokenul JWT.

```
1 usage
public ResponseEntity<AuthenticationResponse> login(AuthenticationRequest request) {
    try {
        authenticationManager.authenticate(
            new UsernamePasswordAuthenticationToken(
                request.getEmail(),
                request.getPassword()
            )
        );
    } catch (BadCredentialsException badCredentialsException) {
        return ResponseEntity.status(HttpStatus.UNAUTHORIZED).body(
            AuthenticationResponse.builder()
                .message("Authentication failed: email or password are wrong")
                .build()
        );
    }
    User user = userRepository.findByEmail(request.getEmail());
    Map<String, Object> claims = new HashMap<>();
    claims.put("role", user.getRole());
    String jwtToken = jwtService.generateToken(claims, user);
    return ResponseEntity.ok(AuthenticationResponse
        .builder()
        .token(jwtToken)
        .build());
}
```

Figura 23: Autentificarea

În figura 23 este prezentată metoda prin care ajunge cererea de la client când utilizatorul încearcă să se autentifice. După verificarea credențialelor, dacă acestea sunt valide, este trimis un token ce conține rolul și alte detalii, cum ar fi data expirării, către client.

În serverul de Angular salvez tokenul în baza de date locală a browserului și am creat un interceptor prin care trece fiecare request și adaugă în antetul cererii tokenul.


```

5+ usages
export const authInterceptor: HttpInterceptorFn = (req : HttpRequest<unknown> , next : HttpHandlerFn ) => {
  const router = inject(Router);
  const token = localStorage.getItem( key: 'studdyBuddy_token');
  if (token) {
    let decodedToken = jwtDecode(token);
    const isExpired = decodedToken.exp ? decodedToken.exp < Date.now() / 1000 : false;
    if (isExpired) {
      localStorage.removeItem( key: 'studdyBuddy_token');
      router.navigate( commands: ['/login']);
      throw new Error('Session expired. Redirecting to login.');
```

Figură 24: Interceptorul pentru adăugarea antetului ce conține tokenul JWT

Figura 24 reprezintă codul sursă pentru adăugarea antetului ce conține tokenul JWT la fiecare cerere pe care o face către server. Tokenul este luat din local storage-ul browserului, se face verificarea, iar dacă tokenul este expirat, utilizatorul este trimis către pagina de login și îi apare mesajul de eroare aferent. Dacă nu este expirat, cererea este clonată și se adaugă tokenul.

5.2 Accesul la baza de date

O alta problema pe care am întâmpinat-o este găsirea celui mai optim mod pentru aplicația mea în legătura cu accesul la date. În Spring Boot am găsit mai multe opțiuni pe care le voi prezenta și voi justifica alegerea făcută.

5.2.1 JDBC Template

JDBC este o interfață care definește cum clienții pot accesa baza de date. Această metodă de a interacționa cu baze de date este orientată către bazele de date relaționale și oferă metode pentru a face cereri, actualizări, cereri și apeluri de proceduri în baza de date.

JdbcTemplate este o clasă centrală în pachetul JDBC care simplifică utilizarea acestuia și ajută la evitarea erorilor comune cum ar fi gestionarea excepțiilor.

Avantaje:

- Mai performant pentru operațiuni de bază.
- Suportă cereri și proceduri stocate.

Dezavantaje:

- Scrierea a mult cod până la executarea unei cereri, cum ar fi crearea conexiunii, crearea cererii, închiderea conexiunii.

- Repetarea codului de la o bază de date la alta implică timp.

5.2.2 MyBatis

MyBatis este un framework de persistență care suportă SQL personalizat. Acesta oferă suport pentru maparea SQL și XML la obiecte Java.

Avantaje:

- Flexibil pentru cereri complexe.
- Este mai flexibil când vine vorba de arhitectura bazei de date.

Dezavantaje:

- MyBatis solicită maparea manuală a câmpurilor obiectelor la rezultatele cererilor SQL.
- Mai puțin orientat pe obiecte.

5.2.3 Spring Data JPA

Spring Data JPA este un modul al framework-ului Spring ce simplifică accesul la baze de date folosind API-ul Java Persistence (JPA). JPA este o specificație Java care descrie o metodologie de mapare a datelor între un model de obiecte și o bază de date relațională. Hibernate este una dintre cele mai populare implementări ale JPA și oferă un set complet de funcționalități pentru managementul persistenței datelor.

Avantaje:

- Simplificare și Reducerea Codului: Elimină necesitatea de a scrie manual codul de acces la date, reducând astfel complexitatea și cantitatea de cod necesar.
- Permite crearea de metode în repository care generează automat interogări pe baza numelor metodologice.
- Suport pentru Query-uri Personalizate: Permite scrierea de query-uri personalizate folosind JPQL (Java Persistence Query Language) sau SQL nativ.
- Caching: Hibernate, ca implementare JPA, oferă mecanisme de caching pentru îmbunătățirea performanței aplicațiilor prin reducerea numărului de interogări la baza de date.

Dezavantaje:

- Abstracție Ridicată: Nivelul de abstractizare poate face dificilă rezolvarea erorilor în comparație cu alte metode.
- Performanță: În unele cazuri, performanța poate fi afectată datorită mecanismelor de gestionare a entităților și a tranzacțiilor.

Astfel, am ales să folosesc JPA în special datorită faptului că face o mapare a datelor între obiecte și baza de date relațională automat cu implementarea Hibernate pentru a adăuga și nivelul de caching pentru performanță. De asemenea, suportul pentru Lazy Loading a fost de

ajutor în contextul în care unele tabele conțin BLOB-uri, aceste date fiind aduse doar când este necesar.

5.2.4 Implementarea JPA

Pentru a configura o aplicație Spring Boot care folosește JPA și Hibernate cu o bază de date MySQL, este nevoie de câteva setări în fișierul `application.properties`.

```
spring.jpa.hibernate.ddl-auto=update
spring.datasource.url=jdbc:mysql://localhost:3306/licenta
spring.datasource.username=root
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
spring.jpa.show-sql= true
```

Figura 25: Configurarea conexiunii cu baza de date

Explicația setărilor:

- `spring.jpa.hibernate.ddl-auto=update`: Această setare specifică strategia Hibernate pentru actualizarea schemei bazei de date. Opțiunea „update” va actualiza schema bazei de date la pornirea aplicației, adăugând tabele noi sau modificând tabelele existente pentru a se potrivi cu modelul obiectului Java.
- Restul setărilor țin de conexiunea cu baza de date.

```
9 usages
@Entity
@NoArgsConstructor
@AllArgsConstructor
@Getter
@Builder
@Setter
public class Enroll {
    no usages
    @Id
    @GeneratedValue(strategy = GenerationType.UUID)
    private String enrollmentId;

    no usages
    @ManyToOne
    private User user;

    no usages
    @ManyToOne
    private Course course;
}
```

Figura 26: Exemplu de entitate

Un repository este o componentă ce oferă un mecanism pentru accesarea, manipularea și interogarea bazelor de date. În continuare voi prezenta un exemplu de repository din aplicație.

```
@Repository
public interface EnrollRepository extends JpaRepository<Enroll, String> {

    1 usage
    @Query("SELECT e.course FROM Enroll e WHERE e.user.email = :email")
    List<Course> findCoursesByUserEmail(@Param("email") String email);

    1 usage
    @Transactional
    @Modifying
    @Query("DELETE FROM Enroll e WHERE e.user.userId = :userId AND e.course.courseId = :courseId")
    int deleteByUserIdAndCourseId(@Param("userId") String userId, @Param("courseId") String courseId);

    1 usage
    void deleteByCourse_CourseId(String courseId);
}
```

Figură 27: Un exemplu de Repository

În repository-ul EnrollRepository observăm că, datorită utilizării Spring Data JPA, metoda deleteByCourse_CourseId(String courseId) este generată automat de bibliotecă pe baza convențiilor de denumire. În schimb, pentru celelalte metode, definesc explicit interogările folosind adnotarea @Query.

5.3 Paginare si căutare după diferite informații

O altă problemă pe care am întâmpinat-o în aplicație a fost paginarea și căutarea. Am folosit paginare pentru lista de exerciții, dar și pentru lista de încărcări a temelor de către studenți.

Problema este că voiam ca aplicația să facă căutarea direct prin cerere în baza de date, nu să fac cerere, apoi să sortez elementele, pentru că acest lucru ar fi redus din performanță. De asemenea, voiam căutarea pe mai multe câmpuri în același timp, ceea ce necesită o cerere complexă.

Pentru a rezolva această problemă, am implementat următoarea soluție:

```
@GetMapping ("/get/assignments")
ResponseBody<?> getAssignments(
    @RequestParam(name = "page", required = false, defaultValue = "0") Long page,
    @ModelAttribute SearchAssignmentsDTO searchAssignmentsDTO) {
    System.out.println(searchAssignmentsDTO);
    long totalRecords = assignmentService.countAssignments(searchAssignmentsDTO);
    long totalPages = (totalRecords + 20 - 1) / 20;
    SearchResultsPageDTO searchResultsPageDTO = new SearchResultsPageDTO();
    List<AssignmentDTO> searchResults = assignmentService.getAllAssignments(searchAssignmentsDTO, page, pageSize: 20L);
    searchResultsPageDTO.setTotalPages(totalPages);
    searchResultsPageDTO.setAssignments(searchResults);
    return ResponseEntity.ok(searchResultsPageDTO);
}
```

Figura 28: Metoda din controller pentru căutarea exercițiilor

1. Expunerea unui REST endpoint care primește de la client numărul paginii și un obiect de tipul SearchAssignmentsDTO care specifică criteriile de căutare.

```
@Getter
@Setter
@ToString
public class SearchAssignmentsDTO {
    no usages
    private String assignmentId;
    no usages
    private String category;
    no usages
    private String title;
    no usages
    private String createdBy;
    no usages
    @DateTimeFormat(pattern = "yyyy-MM-dd")
    private Date createdAt;
}
```

Figura 29: Clasa cu filtrele căutării

2. După cum observăm ne folosim de assignmentService pentru a primi numărul de exerciții care există în baza de date cu aceste criterii.

```
1 usage
public long countAssignments(SearchAssignmentsDTO dto) {
    return assignmentRepository.countAssignmentsByCriteria(dto);
}
```

Figura 30: Metoda pentru numărarea exercițiilor

```
2 usages
@Repository
public interface AssignmentRepository extends JpaRepository<Assignment, String>, CustomAssignmentRepository {
    1 usage
    @Query("SELECT new com.licenta.StuddyBuddy.dto.AssignmentDTO(c.assignmentId, c.category, c.title, c.createdBy, c.createdAt) FROM Assignment c where c.assignmentId = :assignmentId")
    AssignmentDTO getAssignmentMetadata(String assignmentId);

    1 usage
    @Query("SELECT c.content FROM Assignment c where c.assignmentId = :assignmentId")
    byte[] getAssignmentContent(String assignmentId);

    1 usage
    @Query("SELECT c.solution FROM Assignment c where c.assignmentId = :assignmentId")
    byte[] getAssignmentSolution(String assignmentId);
}
```

Figură 31: Repository-ul pentru exerciții

3. Metoda din repository este adusă de extinderea CustomAssignmentRepository.

```

public interface CustomAssignmentRepository {
    1 usage 1 implementation
    List<Assignment> findAssignmentsByCriteria(SearchAssignmentsDTO dto, Long page, Long pageSize);
    1 usage 1 implementation
    long countAssignmentsByCriteria(SearchAssignmentsDTO dto);
}

```

Figura 32: Interfața extensă de AssignmentRepository

5.3.1 Detalii de implementare

Astfel putem observa implementarea metodei în clasa CustomAssignmetRepositroyImpl care implementează această interfața și suprascrie cele două metode.

```

1 usage
@Override
public long countAssignmentsByCriteria(SearchAssignmentsDTO dto) {
    CriteriaBuilder cb = entityManager.getCriteriaBuilder();
    CriteriaQuery<Long> query = cb.createQuery(Long.class);
    Root<Assignment> assignment = query.from(Assignment.class);

    List<Predicate> predicates = new ArrayList<>();

    if (dto.getAssignmentId() != null && !dto.getAssignmentId().isEmpty()) {
        predicates.add(cb.equal(assignment.get("assignmentId"), dto.getAssignmentId()));
    }
    if (dto.getCategory() != null && !dto.getCategory().isEmpty()) {
        predicates.add(cb.equal(assignment.get("category"), dto.getCategory()));
    }
    if (dto.getTitle() != null && !dto.getTitle().isEmpty()) {
        predicates.add(cb.like(cb.lower(assignment.get("title")), pattern: "%" + dto.getTitle().toLowerCase() + "%"));
    }
    if (dto.getCreatedBy() != null && !dto.getCreatedBy().isEmpty()) {
        predicates.add(cb.equal(assignment.get("createdBy"), dto.getCreatedBy()));
    }
    if (dto.getCreatedDate() != null) {
        LocalDate createdDateLocal = dto.getCreatedDate().toInstant().atZone(ZoneId.systemDefault()).toLocalDate();
        predicates.add(cb.equal(cb.function( name: "date", LocalDate.class, assignment.get("createdDate")), createdDateLocal));
    }

    query.select(cb.count(assignment)).where(predicates.toArray(new Predicate[0]));
    return entityManager.createQuery(query).getSingleResult();
}

```

Figura 33: Implementarea funcției pentru căutarea exercițiilor

În această funcție countAssignmentsByCriteria(SearchAssignmentsDTO) se creează un CriteriaBuilder pentru a construi interogări dinamice. CriteriaQuery este configurat pentru a întoarce Long, care reprezintă numărul de înregistrări ce se potrivesc criteriilor. Root reprezintă tabelul de bază, în acest caz Assignment. List<Predicate> este lista ce conține condițiile interogării. Apoi se face un query select care contorizează numărul de exerciții cu predicatele create și returnează rezultatul.

```

int pageSizeValue = (pageSize != null) ? pageSize.intValue() : 20;
int pageNumber = (page != null) ? page.intValue() : 0;
int offset = pageNumber * pageSizeValue;

return entityManager.createQuery(query)
    .setFirstResult(offset)
    .setMaxResults(pageSizeValue)
    .getResultList();

```

Figura 34: Implementarea paginării

Funcția `findAssignmentsByCriteria(SearchAssignmentsDTO, Long page, Long pagesize)` este asemănătoare din punct de vedere al predicatelor, având aceleași criterii de căutare setate, doar că la acest pas este adăugată paginarea prezentă în figura x. Astfel, pentru această cerere trebuie calculat indexul de start pentru interogarea bazei de date și numărul de înregistrări care sunt returnate pe acea pagină, iar acestea sunt date ca parametru în interogare de asemenea.

În final, în controller, după ce acestea sunt setate, este construit un obiect de tipul `SearchResultsPageDTO` care conține numărul total de pagini și lista de exerciții, pentru a fi afișate pe partea de client.

6 STUDIU DE CAZ / EVALUAREA REZULTATELOR

Acest capitol are ca scop principal evaluarea și analiza soluției dezvoltate. Ne vom concentra pe două aspecte esențiale. În primul rând, vom examina dacă soluția funcționează corect, conform specificațiilor definite în capitolul 2. În al doilea rând, vom evalua performanța acesteia, comparativ cu soluțiile existente, utilizând metrice clare și cuantificabile.

Pentru a măsura performanța și corectitudinea aplicației, ne vom folosi de extensia Web Vitals pentru Chrome și de instrumentul Lighthouse. Aceste instrumente ne vor permite să obținem date relevante despre diverse aspecte ale performanței aplicației, inclusiv timpi de încărcare, interactivitate și stabilitate vizuală.

6.1 Analiza Web Vitals

Web Vitals este o inițiativă de la Google ce oferă îndrumare dezvoltatorilor pentru ca aceștia să poată realiza dacă oferă sau nu o experiență bună utilizatorilor.

Web Vitals se focusează pe trei criterii importante:

- LCP (Largest Contentful Paint): Acest criteriu măsoară performanța de încărcare. Pentru o bună experiență, aceasta trebuie să se încadreze între următorii parametri: sub 2.5 secunde experiență bună, între 2.5 și 4s este loc de îmbunătățire, iar ce este peste 4s reprezintă o experiență slabă pentru utilizator.
- INP (Interaction to Next Paint): Criteriu ce măsoară interactivitatea. Pentru ca acest criteriu să se încadreze la o experiență bună, trebuie să fie sub 200ms.

- CLS (Cumulative Layout Shift): Măsoară stabilitatea vizuală (de exemplu, animațiile pe care le are aplicația cât de bine funcționează). Pentru a oferi o experiență bună pentru utilizator, aceasta trebuie să fie mai mică de 0.1.

Astfel, analizând aplicația, am obținut următoarele rezultate:

1. Pe pagina de autentificare, rezultatul analizei ne indică faptul că utilizatorul are parte de o experiență bună, unde locurile de îmbunătățire sunt foarte mici. Putem observa figura 35.

▶ [Web Vitals Extension] LCP 0.528 secs (good)	vitals.js:262
▶ [Web Vitals Extension] FCP 0.528 secs (good)	vitals.js:262
▶ [Web Vitals Extension] TTFB 0.340 secs (good)	vitals.js:262
▶ [Web Vitals Extension] CLS 0.00 (good)	vitals.js:262
▶ [Web Vitals Extension] FID 2 ms (good)	vitals.js:262
▶ [Web Vitals Extension] INP 8 ms (good)	vitals.js:262
▶ [Web Vitals Extension] Interaction 48 ms (good)	vitals.js:262
▶ [Web Vitals Extension] INP 48 ms (good)	vitals.js:262
▶ [Web Vitals Extension] Interaction 16 ms (good)	vitals.js:262
▶ [Web Vitals Extension] Interaction 24 ms (good)	vitals.js:262
▶ [Web Vitals Extension] Interaction 24 ms (good)	vitals.js:262
▶ [Web Vitals Extension] Interaction 16 ms (good)	vitals.js:262
▶ [Web Vitals Extension] Interaction 24 ms (good)	vitals.js:262
▶ [Web Vitals Extension] Interaction 32 ms (good)	vitals.js:262
▶ [Web Vitals Extension] Interaction 32 ms (good)	vitals.js:262
▶ [Web Vitals Extension] Interaction 32 ms (good)	vitals.js:262

Figura 35: Analiză pagină autentificare

2. Pe pagina de administrator, o îmbunătățire posibilă ar fi din perspectiva CLS, mai exact încărcarea problemei la editarea acesteia. Putem observa că indicele CLS este foarte puțin peste metrica noastră de experiență bună a utilizatorului, astfel încât nu este un impact foarte mare.

▶ [Web Vitals Extension] LCP 0.486 secs (good)	VM1073 vitals.js:262
▶ [Web Vitals Extension] LCP 1.187 secs (good)	VM1073 vitals.js:262
▶ [Web Vitals Extension] FCP 0.486 secs (good)	VM1073 vitals.js:262
▶ [Web Vitals Extension] TTFB 0.341 secs (good)	VM1073 vitals.js:262
▶ [Web Vitals Extension] CLS 0.14 (needs-improvement)	VM1073 vitals.js:262
▶ [Web Vitals Extension] Interaction 16 ms (good)	VM1073 vitals.js:262
▶ [Web Vitals Extension] FID 3 ms (good)	VM1073 vitals.js:262
▶ [Web Vitals Extension] INP 16 ms (good)	VM1073 vitals.js:262
▶ [Web Vitals Extension] Interaction 24 ms (good)	VM1073 vitals.js:262
▶ [Web Vitals Extension] Interaction 32 ms (good)	VM1073 vitals.js:262
▶ [Web Vitals Extension] Interaction 40 ms (good)	VM1073 vitals.js:262
▶ [Web Vitals Extension] INP 40 ms (good)	VM1073 vitals.js:262
▶ [Web Vitals Extension] Interaction 24 ms (good)	VM1073 vitals.js:262

Figura 36: Analiza pagina administrator

3. Pe pagina ce arată cursurile la care este înscris atât studentul, cat și profesorul metricile exprimă ca utilizatorul are parte de o experiență bună.

▶ [Web Vitals Extension] LCP 0.504 secs (good)	vitals.js:262
▶ [Web Vitals Extension] LCP 0.771 secs (good)	vitals.js:262
▶ [Web Vitals Extension] LCP 0.838 secs (good)	vitals.js:262
▶ [Web Vitals Extension] FCP 0.504 secs (good)	vitals.js:262
▶ [Web Vitals Extension] TTFB 0.338 secs (good)	vitals.js:262
▶ [Web Vitals Extension] CLS 0.00 (good)	vitals.js:262
▶ [Web Vitals Extension] LCP 0.905 secs (good)	vitals.js:262

Figura 37: Analiză pagina cu toate cursurile la care un utilizator este înrolat

1. Pagina cursurilor de asemenea oferă utilizatorului o experiență bună.

▶ [Web Vitals Extension] LCP 0.490 secs (good)	VM2399 vitals.js:262
▶ [Web Vitals Extension] LCP 0.807 secs (good)	VM2399 vitals.js:262
▶ [Web Vitals Extension] LCP 0.840 secs (good)	VM2399 vitals.js:262
▶ [Web Vitals Extension] FCP 0.490 secs (good)	VM2399 vitals.js:262
▶ [Web Vitals Extension] TTFB 0.349 secs (good)	VM2399 vitals.js:262
▶ [Web Vitals Extension] CLS 0.00 (good)	VM2399 vitals.js:262
▶ [Web Vitals Extension] Interaction 40 ms (good)	VM2399 vitals.js:262
▶ [Web Vitals Extension] FID 2 ms (good)	VM2399 vitals.js:262
▶ [Web Vitals Extension] INP 40 ms (good)	VM2399 vitals.js:262
▶ [Web Vitals Extension] Interaction 32 ms (good)	VM2399 vitals.js:262
▶ [Web Vitals Extension] Interaction 40 ms (good)	VM2399 vitals.js:262
▶ [Web Vitals Extension] Interaction 32 ms (good)	VM2399 vitals.js:262
▶ [Web Vitals Extension] CLS 0.03 (good)	VM2399 vitals.js:262

Figura 38: Analiză pagina cursului

Astfel, analiza Web Vitals arată că majoritatea paginilor din aplicație oferă o experiență bună utilizatorilor. Putem observa că paginile se încadrează în parametrii recomandați de Google pentru o experiență optimă. Cu toate acestea, editarea problemelor poate fi ajustată minor pentru a îmbunătăți stabilitatea vizuală.

6.2 Analiza Lighthouse

Lighthouse este un instrument dezvoltat de Google, asemănător Web Vitals, acesta analizează performanța, accesibilitatea și dacă s-au respectat cele mai bune practici în dezvoltare.

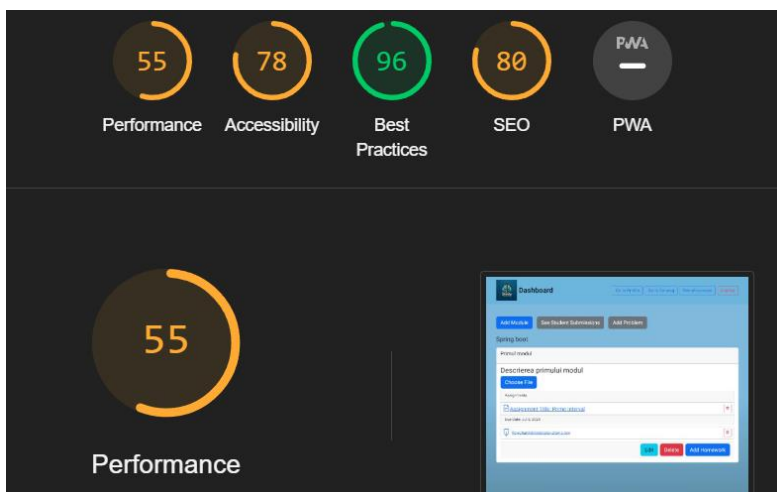


Figura 39: Analiza Light House

Am efectuat o analiză detaliată a paginilor din aplicația noastră, dar ne vom concentra în special pe pagina care va fi cea mai folosită, pagina unui curs.

Rezultatele analizei și locurile de îmbunătățire:

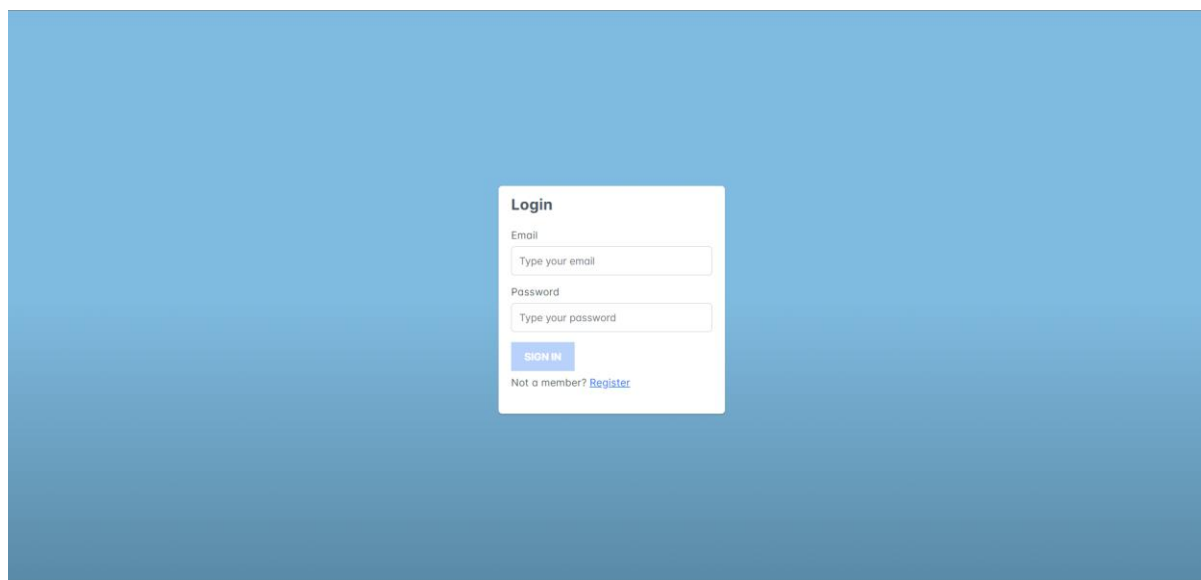
2. Activarea compresiei textului: Activarea compresiei textului poate reduce dimensiunea paginilor web și, astfel, poate îmbunătăți timpii de încărcare.
3. Largest Contentful Paint (LCP) element: Acest aspect se referă la sigla aplicației. Îmbunătățirea timpului de încărcare a acestui element poate reduce LCP și poate îmbunătăți experiența utilizatorilor.
4. Evitarea codului legacy JavaScript: Aceasta se referă la o clasă importată de Angular, care nu este folosită, astfel că nu impactează deloc experiența utilizatorilor.

Rezultatele analizelor efectuate atât cu Web Vitals, cât și cu Lighthouse, indică faptul că aplicația noastră oferă o experiență bună utilizatorilor. Cu toate acestea, există câteva locuri de îmbunătățire. Problemele identificate nu vor impacta semnificativ experiența utilizatorilor, dar optimizările sugerate vor contribui la o experiență și mai bună.

În plus, o viziune pe care o am este migrarea fișierelor la Firebase. Rezultatul acestei schimbări va soluționa problema identificată prin analiza cu aceste instrumente, referitoare la LCP, deoarece Firebase permite o livrare mai rapidă și mai eficientă a resurselor. În concluzie, implementarea acestor optimizări va asigura că aplicația noastră nu doar îndeplinește, ci depășește așteptările utilizatorilor în ceea ce privește performanța și calitatea.

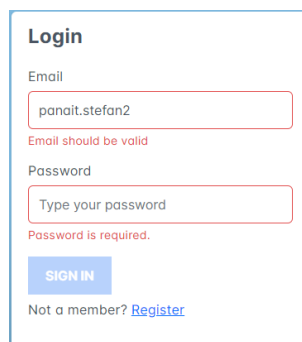
6.3 Evaluarea aplicației

Scopul principal al acestui subcapitol este de a evalua și a analiza soluția dezvoltată. Ne vom concentra pe funcționalitățile aplicației, în special dacă soluția respectă specificațiile definite în capitolele anterioare și funcționează conform așteptărilor.



Figură 40: Autentificare

La accesarea aplicației, utilizatorul este redirecționat către pagina de autentificare. Această pagină conține elemente ce ajută utilizatorul în introducerea datelor.



The screenshot shows a 'Login' form with two input fields. The 'Email' field contains 'panait.stefan2' and has a red border with the error message 'Email should be valid' below it. The 'Password' field contains 'Type your password' and has a red border with the error message 'Password is required.' below it. A blue 'SIGN IN' button is at the bottom, and a link 'Not a member? Register' is below the button.

Figura 41: Erorile paginii de autentificare

Astfel, dacă utilizatorul nu introduce un email corect sau parola nu este completată, acesta este notificat cu erori.

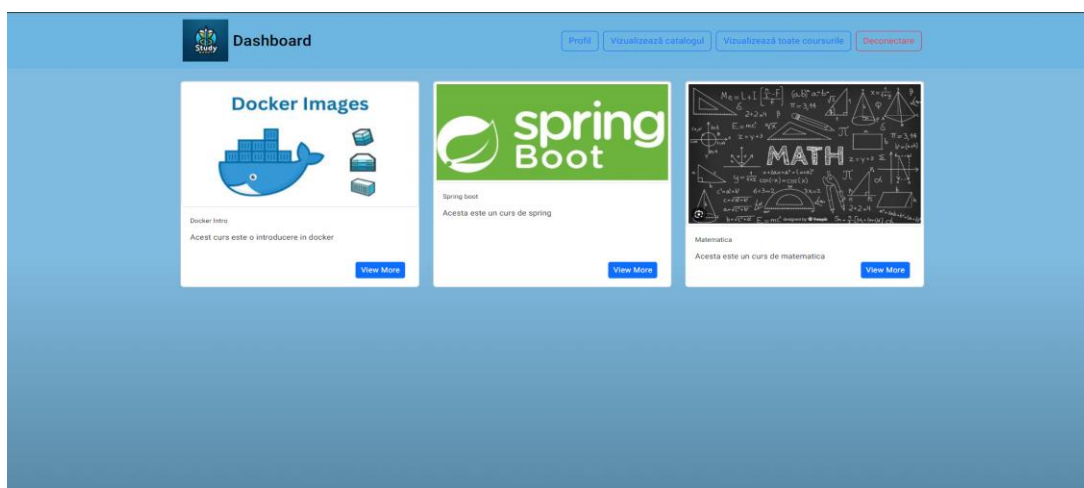


Figura 42: Pagina principală

După autentificare, utilizatorul este redirecționat către pagina principală unde poate vizualiza cursurile la care este înscris. De asemenea, utilizatorul poate vizualiza bara de navigație de unde poate accesa profilul, catalogul sau toate cursurile existente.

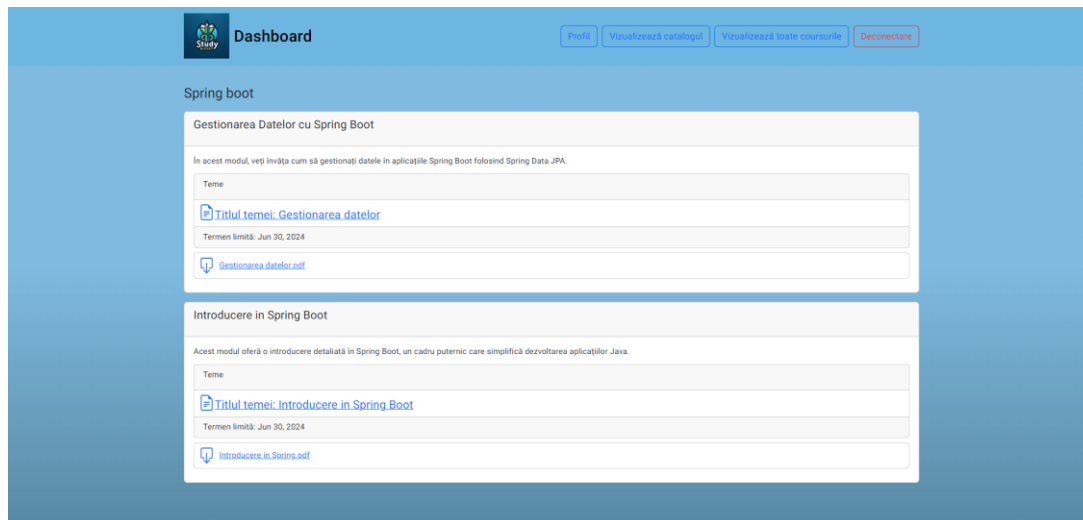


Figura 43: Pagina unui curs

Atunci când utilizatorul intră pe un curs, acesta poate vizualiza modulele cursului, titlul, descrierea, temele și suportul de curs atașat modulelor.

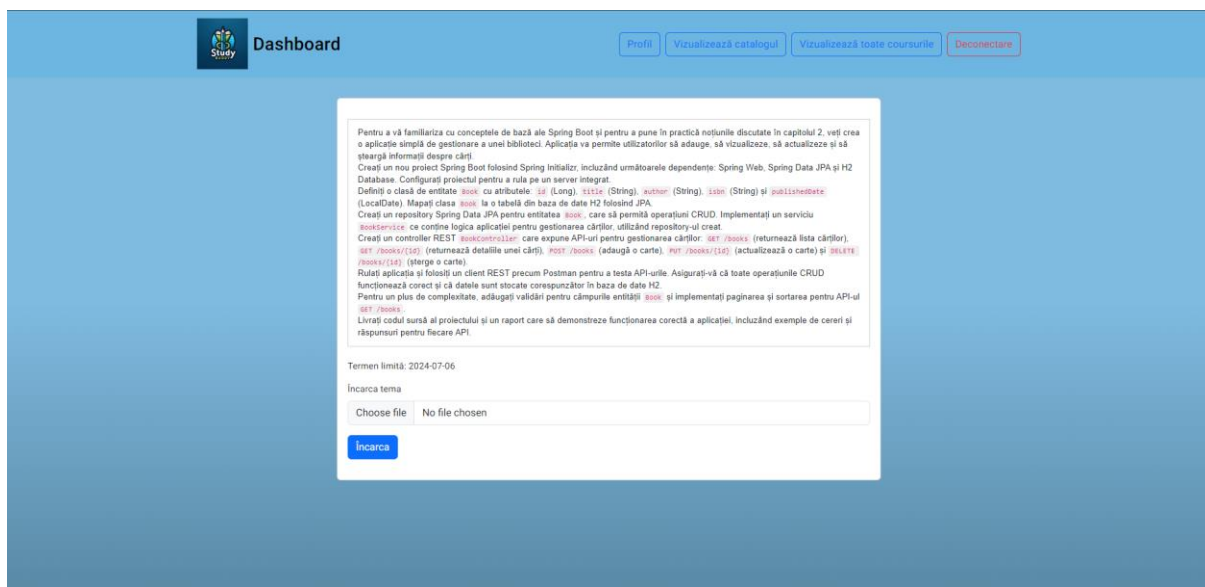


Figura 44: Vizualizarea unei teme

La deschiderea unei teme, o pagina cu cerința și posibilitatea de a încărca tema sunt vizibile.

Pentru a vă familiariza cu conceptele de bază ale Spring Boot și pentru a pune în practică noțiunile discutate în capitolul 2, veți crea o aplicație simplă de gestionare a unei biblioteci. Aplicația va permite utilizatorilor să adauge, să vizualizeze, să actualizeze și să șteargă informații despre cărți.

Creați un nou proiect Spring Boot folosind Spring Initializr, incluzând următoarele dependențe: Spring Web, Spring Data JPA și H2 Database. Configurați proiectul pentru a rula pe un server integrat.

Definiți o clasă de entitate `Book` cu atributele: `id` (Long), `title` (String), `author` (String), `isbn` (String) și `publishedDate` (LocalDate). Mapați clasa `Book` la o tabelă din baza de date H2 folosind JPA.

Creați un repository Spring Data JPA pentru entitatea `Book`, care să permită operațiuni CRUD. Implementați un serviciu `BookService` ce conține logica aplicației pentru gestionarea cărților, utilizând repository-ul creat.

Creați un controller REST `BookController` care expune API-uri pentru gestionarea cărților: `GET /books` (returnează lista cărților), `GET /books/{id}` (returnează detaliile unei cărți), `POST /books` (adaugă o carte), `PUT /books/{id}` (actualizează o carte) și `DELETE /books/{id}` (șterge o carte).

Rulați aplicația și folosiți un client REST precum Postman pentru a testa API-urile. Asigurați-vă că toate operațiunile CRUD funcționează corect și că datele sunt stocate corespunzător în baza de date H2.

Pentru un plus de complexitate, adăugați validări pentru câmpurile entității `Book` și implementați paginarea și sortarea pentru API-ul `GET /books`.

Livrați codul sursă al proiectului și un raport care să demonstreze funcționarea corectă a aplicației, incluzând exemple de cereri și răspunsuri pentru fiecare API.

Termen limită: 2024-07-06

Ultima încărcare a temei: tema.zip

[Descarca tema](#)

Încarca tema

[Încarca](#)

Figura 45: Vizualizare descărcare temă

De altfel dacă o temă a fost încărcată anterior, studentul are posibilitatea de a descărca tema și vizualiza ce a încărcat.

Nota:

10

Recenzia temei:

Bravo ai utilizat excelent tehnicile invatate in acest modul

Rezolvarea corectă

[GitHub: Let's build from here · GitHub](#)

Figura 46: Recenzia profesorului

Dacă temă a fost corectată de către un profesor, elevul, pe aceeași pagină unde a încercat sau în catalog, poate citi recenzia, observa nota și accesa rezolvarea corectă a temei.

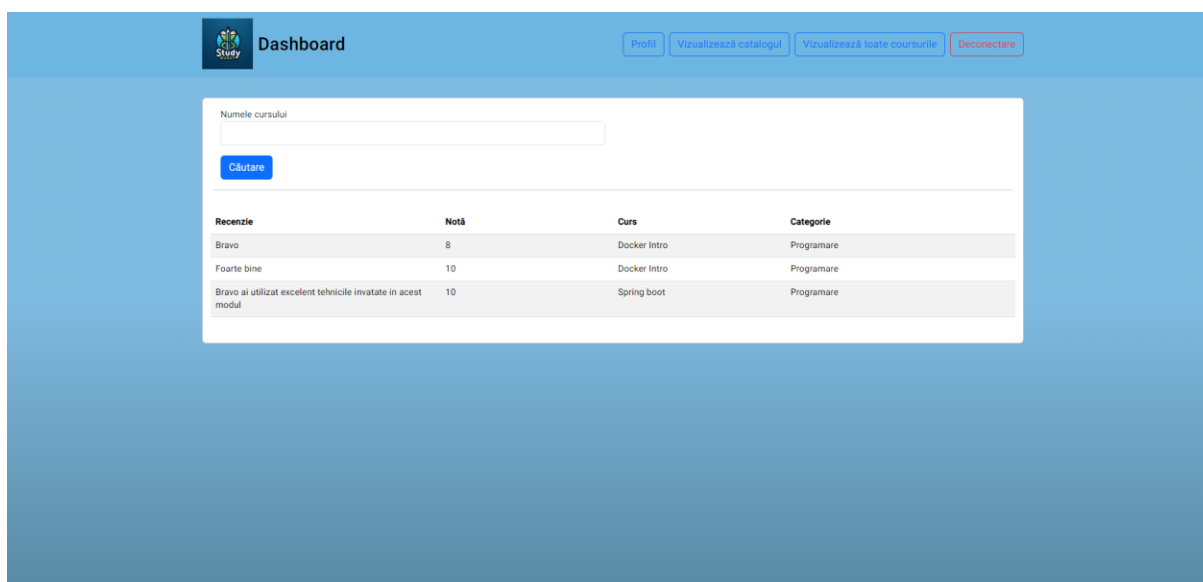


Figura 47: Vizualizare catalog

Pagina catalogului dispune, de asemenea, de căutare în funcție de numele unui curs, pentru o utilizare mai prietenoasă.



Figura 48: Buton editare siglă

Diferența dintre pagina principală a profesorului și cea a studentului este că profesorul are și butonul de editare, de unde poate edita sigla cursului.

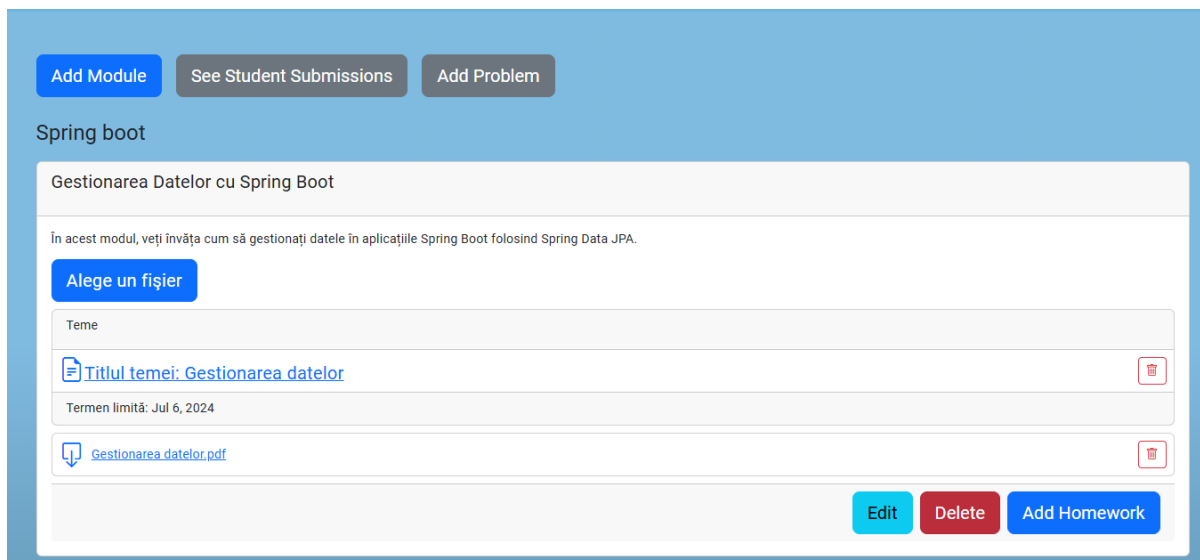


Figura 49: Vizualizarea cursului în rolul de profesor

De asemenea, pagina cursului este similară pentru profesor, cu diferite excepții, acesta având dreptul de a adăuga, șterge și edita module, adăuga și șterge teme, încărca și șterge suportul de curs.

Add Module

Title

Description

Add Module

Cancel

Figura 50: Adaugarea modulului

Search Submissions

Assignment Instance ID:

Assignment Name:

Student First Name:

Student Last Name:

Submission Date:

mm/dd/yyyy

Search

Submissions




Assignment Name	Submission id	Student First Name	Student Last Name	Submission Date	File	Grade	Actions
Graf neorientat	182d1892-b290-494c-a26c-53365ea5f432	Bogdan	Vasile	2024-05-31 12:13	 Panait_Stefan_Andrei_342C2 (5).zip	8	<div>Review</div>
Ecuația de gradul 1	ba2290ce-49ec-49db-b0a3-744e11a6d92e	Bogdan	Vasile	2024-05-31 15:33	 Panait_Stefan_Andrei_342C2 (5).zip	10	<div>Review</div>
Gestionarea datelor	d134ef58-3c2b-4967-9607-e7b0d1707c63	Bogdan	Vasile	2024-06-14 13:58	 tema.zip	10	<div>Review</div>

Figura 51: Vizualizarea temelor studenților

Dacă profesorul apasă pe "vizualizează încărcările studenților", acestuia i se deschide pagina din figura 51, unde poate descărca ceea ce elevul a încărcat. De asemenea, poate căuta după anumite criterii, cum ar fi numele studentului, data încărcării, numele temei, ID-ul temei.

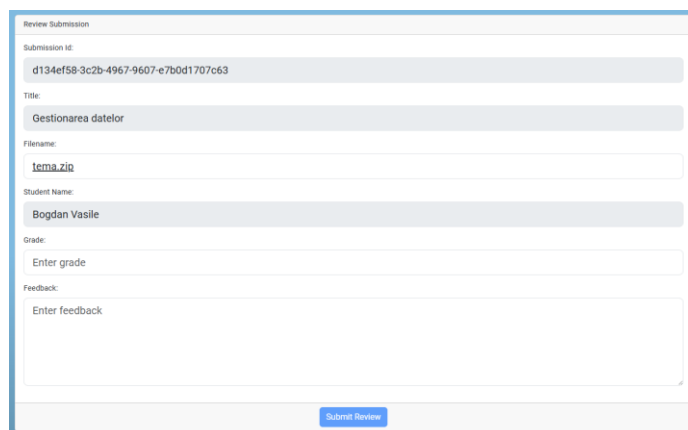


Figura 52: Adaugarea unei recenzii

Prin apăsarea butonului de "Review", profesorului îi va apărea pe ecran un formular unde poate completa nota, dar și o recenzie asupra temei.

Dacă profesorul apasă pe butonul "Adaugă o problemă", acestuia i se deschide un formular unde poate completa titlul, cerința, rezolvarea corectă și categoria în care se încadrează tema. Pentru rezolvarea corectă și cerință, profesorul are posibilitatea de a adăuga poze și link-uri datorită Rich Text Editor-ului. Această funcționalitate este disponibilă și administratorului.

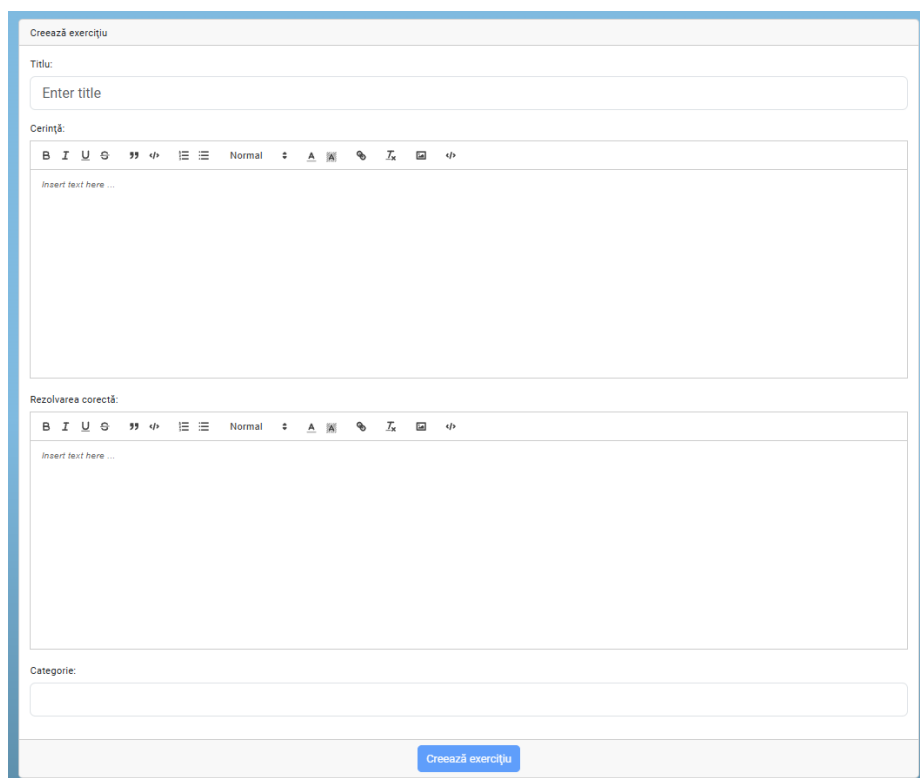


Figura 53: Adaugarea unui exercițiu

Exerciții:

Assignment ID:
 Categorie:
 Creat de:

Titlu:
 Data creării:

Assignment ID	Titlu	Creat de	Data creării	Categorie	Acțiuni
0f070e67-49ba-427b-9e90-187886043808	Exerciții polinoame	panait.stefan2112@gmail.com	2024-05-31	Matematica	<input type="checkbox"/> <input type="checkbox"/> <input type="button" value="Select"/>
132d9991-b058-4486-af93-9d56f8ac087c	Gestionarea datelor	dani@gmail.com	2024-06-14	Spring Boot	<input type="checkbox"/> <input type="checkbox"/> <input type="button" value="Select"/>
1b70b5d4-ad39-44d2-8567-4f727f6d7cc	Exista 3 numere pozitive	panait.stefan2112@gmail.com	2024-05-31	Algoritmica	<input type="checkbox"/> <input type="checkbox"/> <input type="button" value="Select"/>
2851e163-d474-430b-994b-6e6b7aa6f615	Subiect evaluarea nationala	panait.stefan2112@gmail.com	2024-05-31	Romana	<input type="checkbox"/> <input type="checkbox"/> <input type="button" value="Select"/>
2c983552-b23a-466f-93f5-825de733625d	Ecuatia de gradul 2	panait.stefan2112@gmail.com	2024-05-31	Algoritmica	<input type="checkbox"/> <input type="checkbox"/> <input type="button" value="Select"/>
2cdc1594-d884-4850-9ccc-2e609c238c54	Exerciții trigonometrie	panait.stefan2112@gmail.com	2024-05-31	Matematica	<input type="checkbox"/> <input type="checkbox"/> <input type="button" value="Select"/>
4334f6b7-d015-4366-af90-ce4bfc0f8d8f	Graf neorientat	panait.stefan2112@gmail.com	2024-05-30	Algoritmica	<input type="checkbox"/> <input type="checkbox"/> <input type="button" value="Select"/>
585d4b1e-4930-4186-ad0e-4603eaa01a76	CMMC	panait.stefan2112@gmail.com	2024-05-31	Algoritmica	<input type="checkbox"/> <input type="checkbox"/> <input type="button" value="Select"/>
70fdb1cf-da51-480d-9911-d616a52eaa9	Prime interval	panait.stefan2112@gmail.com	2024-05-31	Algoritmica	<input type="checkbox"/> <input type="checkbox"/> <input type="button" value="Select"/>
d39cf824-d497-458b-bc62-6899b0bef492	Introducere in Spring Boot	dani@gmail.com	2024-06-14	test	<input type="checkbox"/> <input type="checkbox"/> <input type="button" value="Select"/>
dea7b954-5b7a-458a-abac-239968512c53	Ecuatia de gradul 1	panait.stefan2112@gmail.com	2024-05-31	Algoritmica	<input type="checkbox"/> <input type="checkbox"/> <input type="button" value="Select"/>
e286782e-bd52-4f28-bf33-4bdf87c42860	Exerciții logaritmi	panait.stefan2112@gmail.com	2024-05-31	Matematica	<input type="checkbox"/> <input type="checkbox"/> <input type="button" value="Select"/>
ea0e7cef-ba44-470f-9364-09ee558f9fe0	Verifica numar prim	panait.stefan2112@gmail.com	2024-05-31	Algoritmica	<input type="checkbox"/> <input type="checkbox"/> <input type="button" value="Select"/>
18ba30fb-4ae6-434f-a70b-27bf524360f5	Exerciții numere complexe	panait.stefan2112@gmail.com	2024-05-31	Matematica	<input type="checkbox"/> <input type="checkbox"/> <input type="button" value="Select"/>

Termen limită:

Figura 54: Adăugarea unei teme

Apăsând pe butonul “Adaugă temă”, profesorului îi apare lista de exerciții, acesta putând căuta după anumite criterii un exercițiu. Acesta poate alege unul dintre exerciții și selecta un termen limită pentru temă, apoi, apăsând butonul “Adaugă temă”, acesta postează tema pentru a fi vizibilă studenților.

Creează un curs

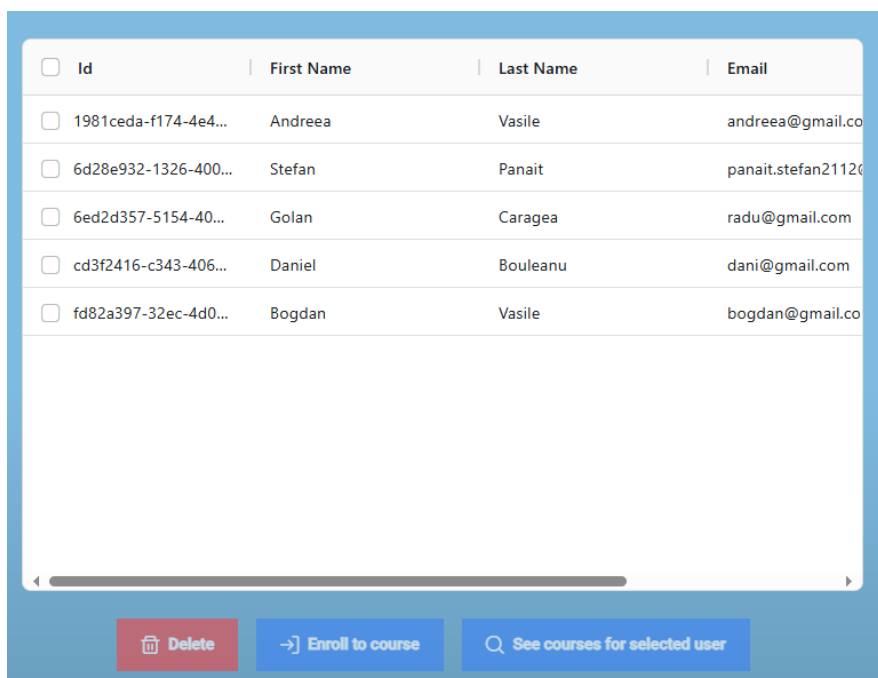
Numele cursului:

Descriere:

Categorie:

Figura 55: Pagina pentru utilizatorii cu rolul de administrator

După autentificarea utilizatorului care are rolul de administrator, acestuia i se deschide pagina din figura 55. De aici, administratorul poate adăuga cursuri, adăuga utilizatori, administra utilizatorii, adăuga exerciții, administra lista de exerciții și administra cursurile.



<input type="checkbox"/>	Id	First Name	Last Name	Email
<input type="checkbox"/>	1981ceda-f174-4e4...	Andreea	Vasile	andreea@gmail.co
<input type="checkbox"/>	6d28e932-1326-400...	Stefan	Panait	panait.stefan2112@
<input type="checkbox"/>	6ed2d357-5154-40...	Golan	Caragea	radu@gmail.com
<input type="checkbox"/>	cd3f2416-c343-406...	Daniel	Bouleanu	dani@gmail.com
<input type="checkbox"/>	fd82a397-32ec-4d0...	Bogdan	Vasile	bogdan@gmail.co

Figura 56: Tabel ce conține toti utilizatorii

Pentru administrarea utilizatorilor, administratorul vizualizează un tabel și poate înscrie un utilizator la un curs, poate modifica detaliile utilizatorilor, poate șterge un utilizator din sistem sau de la un curs. De asemenea, pagina de administrare a cursurilor este asemănătoare cu această pagină.

De asemenea, administratorul și profesorul pot edita sau șterge exerciții din lista de exerciții. La editarea unui exercițiu, acestuia îi sunt completate automat informațiile exercițiului din baza de date pentru a putea fi editat.

6.4 Dezvoltări viitoare

În acest subcapitol, vom explora câteva dintre dezvoltările viitoare pe care intenționez să le implementez. Voi explica de ce consider că aceste dezvoltări contribuie la îmbunătățirea aplicației din punct de vedere al cerințelor nonfuncționale, dar și al celor funcționale.

6.4.1 Migrarea fișierelor către un sistem ca Firebase

Firebase este o platformă dezvoltată de Google ce vine în ajutorul dezvoltatorilor pentru crearea aplicațiilor. componentele principale Firebase sunt baze de date, stocare de fișiere, hosting, funcții cloud și multe altele.

Migrarea fișierelor către un sistem precum Firebase este benefică din mai multe motive

1. Poate gestiona fișiere de toate dimensiunile și volume mari de date fără probleme, adaptându-se automat la nevoile aplicației tale.
2. Utilizează reguli de securitate Firebase pentru a proteja fișierele, permițându-ți să configurezi cine poate accesa și încărca fișiere.

3. Fișierele sunt distribuite global, ceea ce reduce latența și îmbunătățește timpul de acces pentru utilizatori.
4. Este ușor de integrat în aplicațiile tale mobile și web prin SDK-urile Firebase, oferind metode simple pentru încărcarea și descărcarea fișierelor.

6.4.2 Crearea unui nou rol de utilizator

O altă dezvoltare pe care doresc să o implementez pe viitor în aplicație este crearea unui nou rol de utilizator, mai exact rolul de părinte. Astfel, părintele ar avea acces la notele și recenziile profesorilor și ar putea primi actualizări legate de evoluția studentului.

7 CONCLUZII

În concluzie, consider că aplicația dezvoltată îndeplinește obiectivele propuse în capitolele incipiente, fiind o aplicație ce va contribui la dezvoltarea din punct de vedere educațional. Aplicația nu doar că atinge scopurile funcționale propuse, dar și satisface cerințele nonfuncționale în ceea ce privește securitatea, fiabilitatea și performanța.

De asemenea, este important de menționat că, deși educația online oferă numeroase avantaje, nu sunt adeptul educației exclusive online. Consider că întâlnirea fizică dintre studenți și profesori, dar chiar și dintre studenți și studenți, este imperativă și contribuie foarte mult la dezvoltarea individului. Astfel, aplicația îndeplinește scopul propus, aceasta fiind o extensie valoroasă a învățării tradiționale.

Realizarea acestor obiective este un pas major înainte, dar mereu există loc pentru îmbunătățiri și dezvoltări ulterioare. Aplicația va continua să evolueze, integrând feedback-ul utilizatorilor și adaptându-se la noile cerințe și tehnologii, pentru a rămâne un instrument relevant și eficient în procesul educațional.

8 BIBLIOGRAFIE

- [1] Diana Gimiși, "Piața meditațiilor, estimată la peste 120 de milioane de lei pe lună, pe baza datelor publice și a cercetărilor recente – analiză" [Online]. Available: <https://www.edupedu.ro/piata-meditatiilor-estimata-la-pest-120-de-milioane-de-lei-pe-luna-pe-baza-datelor-publice-si-a-cercetarilor-recente-analiza/>
- [2] Melike Ömeroğulları, Karin Guill, Olaf Köller, "Effectiveness of private tutoring during secondary schooling in Germany: Do the duration of private tutoring and tutor qualification affect school achievement?" Preluat: <https://www.sciencedirect.com/science/article/pii/S0959475219303093>

- [3] Rai News it "Scuola, 1 studente su 5 va a ripetizioni. Così le famiglie spendono circa 450 euro" Preluat: rainews.it
- [4] Documentație Spring Boot Preluat: <https://spring.io/projects/spring-boot>
- [5] Documentație Angular Preluat: <https://angular.dev/overview>
- [6] Documentație MySQL Preluat: <https://dev.mysql.com/doc/>
- [7] Documentație Docker Preluat: <https://docs.docker.com/>
- [8] Documentație JSON Web Tokens Preluat: <https://jwt.io/introduction>
- [9] Documentație MyBatis Preluat: <https://mybatis.org/mybatis-3/getting-started.html>
- [10] Documentație Spring Data JDBC Preluat: <https://spring.io/projects/spring-data-jdbc>
- [11] Documentație Spring Data JPA Preluat: <https://spring.io/projects/spring-data-jpa>
- [12] Geeksforgeeks, "Angular vs ReactJS : Which one to Choose as Frontend Development Framework in 2024", Preluat: <https://www.geeksforgeeks.org/angular-vs-reactjs/>